Kalyanmoy Deb
Kaisa Miettinen
Shamik Chaudhuri

An Estimation of Nadir Objective Vector Using a Hybrid Evolutionary-cum-Local-Search Procedure

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS

W-470

Kalyanmoy Deb – Kaisa Miettinen – Shamik Chaudhuri

An Estimation of Nadir Objective Vector Using a Hybrid Evolutionary-cum-Local-Search Procedure

Business Technology

May
2009

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS
WORKING PAPERS
W-470

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS
PL 1210
FI-00101 HELSINKI
FINLAND

ISSN 1235-5674
(Electronic working paper)
ISBN 978-952-488-326-9

Helsinki School of Economics -
HSE Print 2009

An Estimation of Nadir Objective Vector Using a Hybrid Evolutionary-cum-Local-Search Procedure

Kalyanmoy Deb*, Kaisa Miettinen†, and Shamik Chaudhuri‡

Abstract

A nadir objective vector is constructed from the worst Pareto-optimal objective values in a multi-objective optimization problem and is an important entity to compute because of its significance in estimating the range of objective values in the Pareto-optimal front and also in executing a number of interactive multi-objective optimization techniques. Along with the ideal objective vector, it is also needed for the purpose of normalizing different objectives, so as to facilitate a comparison and agglomeration of the objectives. However, the task of estimating the nadir objective vector necessitates information about the complete Pareto-optimal front and has been reported to be a difficult task, and importantly an unsolved and open research issue. In this paper, we propose certain modifications to an existing evolutionary multi-objective optimization procedure to focus its search towards the extreme objective values and combine it with a reference-point based local search approach to constitute a couple of hybrid procedures for a reliable estimation of the nadir objective vector. With up to 20-objective optimization test problems and on a three-objective engineering design optimization problem, one of the proposed procedures is found to be capable of finding the nadir objective vector reliably. The study clearly shows the significance of an evolutionary computing based search procedure in assisting to solve an age-old important task in the field of multi-objective optimization.

Keywords: Nadir point, multi-objective optimization, non-dominated sorting GA, evolutionary multi-objective optimization (EMO), multiple objectives, hybrid procedure, ideal point, Pareto optimality.

1 Introduction

In a multi-objective optimization procedure, the estimation of a nadir objective vector (or simply a nadir point) is often an important task. The nadir objective vector is constructed from the worst values of each objective function corresponding to the entire set of Pareto-optimal solutions, that is, the Pareto-optimal front. Sometimes, this point is confused with the point representing the worst objective values of the entire search space, which is often an over-estimation of the true nadir objective vector. The importance of finding the nadir objective vector was recognized by the multiple criteria decision making (MCDM) researchers and practitioners since early seventies. However, even after about 40 years of active research in multi-objective optimization and decision making, there does not exist a reliable procedure of finding the nadir point in problems having more than three objectives. For this reason, a reliable estimation of the nadir point is an important matter to anyone interested in multi-objective optimization, including evolutionary multi-objective optimization (EMO) researchers and practitioners. We outline here the motivation and need for finding the nadir point.

1. Along with the ideal objective vector (a point constructed from the best values of each objective), the nadir objective vector can be used to *normalize* objective functions [30], a matter often desired for an adequate functioning of multi-objective optimization algorithms in the presence of objective functions with different magnitudes. With these two extreme values, the objective functions can be scaled so that each scaled objective takes values more or less in the same range. These scaled values can be used for optimization with different algorithms like the reference point method, weighting method, compromise programming, the Tchebycheff method (see [30] and references therein), or even for EMO algorithms. Such a scaling procedure may help in

*Kalyanmoy Deb holds the Finland Distinguished Professorship at Department of Business Technology, Helsinki School of Economics, PO Box 1210, FI-00101 Helsinki, Finland (Kalyanmoy.Deb@hse.fi) and Deva Raj Chair Professorship at Department of Mechanical Engineering, Indian Institute of Technology Kanpur, PIN 208016, India (deb@iitk.ac.in).

†Kaisa Miettinen is a Professor at the Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland (kaisa.miettinen@jyu.fi). Earlier, she worked at Helsinki School of Economics, Finland.

‡Shamik Chaudhuri is currently working in John F. Welch Technology Centre, Plot 122, Export Promotion Industrial Park Phase 2, Hoodi Village, Whitefield Road, Bangalore, PIN 560066, India (shamikc@gmail.com).

reducing the computational cost by solving the problem faster [34].

2. The second motivation comes from the fact that the nadir objective vector is a pre-requisite for finding preferred Pareto-optimal solutions in different interactive algorithms, such as the *guess* method [5] (where the idea is to maximize the minimum weighted deviation from the nadir objective vector), or it is otherwise an integral part of an interactive method like the NIMBUS method [30, 33]. The knowledge of a nadir point should also help in interactive EMO procedures, one implementation of which has been suggested recently [14] and many other possibilities are discussed in [3].

3. Thirdly, the knowledge of nadir and ideal objective values helps the decision-maker in adjusting her/his expectations on a realistic level by providing the range of each objective and can then be used to aid in specifying preference information in interactive methods in order to focus on a desired region of the Pareto-optimal front.

4. Fourthly, in visualizing a Pareto-optimal front, the knowledge of the nadir objective vector is crucial. Along with the ideal point, the nadir point provides the range of each objective in order to facilitate comparison of different Pareto-optimal solutions, that is, visualizing the trade-off information through value paths, bar charts, petal diagrams etc. [30, 31].

5. Above all, the task of accurately estimating the nadir point in a three or more objective problem is a non-trivial and challenging task, and is an open research topic till to date. Researchers have repeatedly shown that the task is difficult even for linear multi-objective optimization problems. Therefore, any new effort to arrive at a suitable methodology for estimating the nadir point has an intellectual and pedagogic importance, despite its practical significance outlined above.

These motivations for estimating the nadir point led the researchers dealing with MCDM methodologies to suggest procedures for approximating the nadir point using a so-called *payoff table* [1]. This involves computing individual optimum solutions for objectives, constructing a payoff table by evaluating other objective values at these optimal solutions, and estimating the nadir point from the worst objective values from the table. This procedure may not guarantee a true estimation of the nadir point for more than two objectives. Moreover, the estimated nadir point can be either an over-estimation or an under-estimation of the true nadir point. For example, Iserman and Steuer [23] have demonstrated these difficulties for finding a nadir point using the payoff table method even for linear problems and emphasized the need of using a better method. Among others, Dessouky et al. [20] suggested three heuristic methods and Korhonen et al. [27] another heuristic method for this purpose. Let us point out that all these methods suggested have been developed for linear multi-objective problems where all objectives and constraints are linear functions of the variables.

In [21], an algorithm for deriving the nadir point is proposed based on subproblems. In other words, in order to find the nadir point for an M-objective problem, Pareto-optimal solutions of all $\binom{M}{2}$ bi-objective optimization problems must first be found. Such a requirement may make the algorithm computationally impractical beyond three objectives, although Szczepanski and Wierzbicki [37] implemented the above idea using evolutionary algorithms (EAs) and showed successful applications with up to four objective linear optimization problems. Moreover, authors of [21] did not suggest how to realize the idea in non-linear problems. It must be emphasized that because the determination of the nadir point depends on finding the worst objective values in the set of Pareto-optimal solutions, even for linear problems, this is a difficult task [2].

Since an estimation of the nadir objective vector necessitates information about the whole Pareto-optimal front, any procedure of estimating this point should involve finding Pareto-optimal solutions. This makes the task more difficult compared to finding the ideal point [27]. Since EMO algorithms can be used to find a representation of the entire or a part of the Pareto-optimal front, EMO methodologies stand as viable candidates for this task. Another motivation for using an EMO procedure is that nadir point estimation is to be made only once in a problem at the beginning of the decision making process before any human decision maker is included. So, even if the proposed procedure uses somewhat substantial computational effort (one of the criticisms made often against evolutionary optimization methods), a reliable and accurate methodology for estimating the nadir point is desired.

A careful thought will reveal that an estimation of the nadir objective vector may not need finding the complete Pareto-optimal front, but only an adequate number of *critical* Pareto-optimal solutions may be enough for this task. Based on this concept, an earlier preliminary study by the authors [15] showed that by altering the usual definition of a crowding distance metric of an existing EMO methodology (elitist non-dominated sorting GA or NSGA-II [13]) to emphasize objective-wise best and worst Pareto-optimal solutions (we call here as *extreme* solutions), a near nadir point can be estimated on a number of test problems. Since this study, we realized that the proposed NSGA-II procedure alone was not enough to find the desired extreme solutions in a finite amount of computational effort, when applied to other more tricky optimization problems. In this paper, we hybridize the previously proposed NSGA-II approach with a local search procedure which uses the idea of an achievement scalarizing function utilized, for example, in an interactive MCDM approach – the reference point approach [38] – to enhance the convergence of solutions to the desired extreme points. This extension, by far, is not an easy task, as a local search in any form in the context of multiple conflicting objectives is a difficult proposition. empirical results of this hybrid nadir point estimation procedure on problems with up to 20 objectives, on some difficult numerical optimization problems, and on an engineering design problem amply demonstrate the usefulness and promise of the proposed hybrid procedure.

The rest of this paper is organized as follows. In Section II, we introduce basic concepts of multi-objective optimization and discuss the importance and difficulties of estimating the nadir point. In Section III, we describe two modified NSGA-II approaches for finding near extreme Pareto-optimal solutions. The nadir point estimation procedures proposed based on a hybrid evolutionary-cum-local-search concept are described in Section IV. The performances of the modified NSGA-II procedures are tested and compared with a naive approach on a number of scalable numerical test problems and the results are described in Section V. The use of the hybrid nadir point estimation procedure in full is demonstrated in Section VI by solving three test problems, including an engineering design problem. Some discussions and possible extensions of the study are presented in Section VII. Finally, the paper is concluded in Section VIII.

2 Nadir Objective Vector and Difficulties of its Estimation

We consider multi-objective optimization problems involving M conflicting objectives ($f_i : \mathcal{S} \rightarrow \mathbf{R}$) as functions of decision variables $\mathbf{x}$:

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in \mathcal{S}, \end{array} \tag{1}$$

where $\mathcal{S} \subset \mathbf{R}^n$ denotes the set of feasible solutions. A vector consisting of objective function values calculated at some point $\mathbf{x} \in \mathcal{S}$ is called an objective vector $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))^T$. Problem (1) gives rise to a set of *Pareto-optimal* solutions or a Pareto-optimal front (P^*), providing a trade-off among the objectives. In the sense of minimization of objectives, Pareto-optimal solutions can be defined as follows [30]:

Definition 1 *A decision vector $\mathbf{x}^* \in \mathcal{S}$ and the corresponding objective vector $\mathbf{f}(\mathbf{x}^*)$ are* Pareto-optimal *if there does not exist another decision vector $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, 2, \ldots, M$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one index j.*

Let us mention that if an objective f_j is to be maximized, it is equivalent to minimize $-f_j$. In what follows, we assume that the Pareto-optimal front is bounded. We now define a *critical* point, as follows:

Definition 2 *A point $\mathbf{z}^{(j)^c}$ is a* critical *point with respect to the j-th objective function, if it corresponds to the worst value of f_j among all Pareto-optimal solutions, i.e., $\mathbf{z}^{(j)^c} = \{\mathbf{f}(y) | y = \text{argmax}_{\mathbf{x} \in P^*} f_j(\mathbf{x})\}$.*

The nadir objective vector can now be defined as follows.

Definition 3 *An objective vector $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \ldots, z_M^{\text{nad}})^T$ whose j-th element is taken from j-th component of the corresponding critical Pareto-optimal point $z_j^{\text{nad}} = z_j^{(j)^c}$ is called a* nadir *objective vector.*

Due to the requirement that the a critical point must be an Pareto-optimal point, the estimation of the nadir objective vector is, in general, a difficult task. Unlike the *ideal objective vector* $\mathbf{z}^* = (z_1^*, z_2^*, \ldots, z_M^*)^T$, which can be found by minimizing each objective individually over the feasible set S (i.e., $z_j^* = \min_{\mathbf{x} \in \mathcal{S}} f_j(\mathbf{x})$), the nadir point cannot be formed by maximizing objectives individually over $\mathcal{S}$. To find the nadir point, Pareto-optimality of solutions used for constructing the nadir point must be first established. This makes the task of finding the nadir point a difficult one.

To illustrate this aspect, let us consider a bi-objective minimization problem shown in Figure 1. If we maximize f_1 and f_2 individually, we obtain points A and B, respectively. These two points can be used to construct the so-called *worst objective vector*, $\mathbf{z}^w$. In many problems (even in bi-objective optimization problems), the nadir objective vector and the worst objective vector are not the same point, which can also be seen in Figure 1. In order to estimate the nadir point correctly, we need to find critical points (such as C and D in Figure 1).

2.1 Payoff Table Method

Benayoun et al. [1] introduced the first interactive multi-objective optimization method and used a nadir point (although the authors did not use the term 'nadir'), which was to be found by using a payoff table. To be more specific, each objective function is first minimized individually and then a table

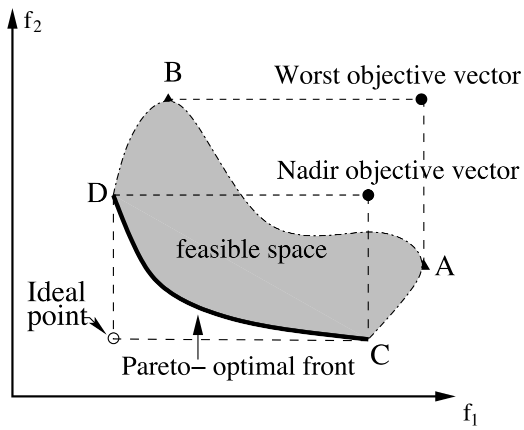

Figure 1: The nadir and worst objective vectors may be different.

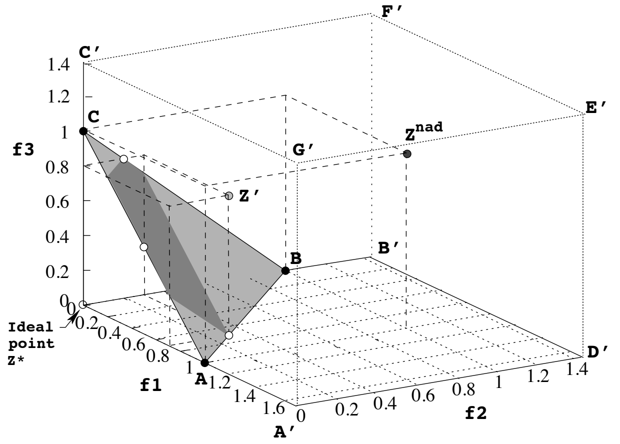

Figure 2: Payoff table may not produce the true nadir point.

is constructed where the i-th row of the table represents values of all objective functions calculated at the point where the i-th objective obtained its minimum value. Thereafter, the maximum value of the j-th column can be considered as an estimate of the upper bound of the j-th objective in the Pareto-optimal front and these maximum values together may be used to construct an approximation of the nadir objective vector. The main difficulty of such an approach is that solutions are not necessarily unique and thus corresponding to the minimum solution of an objective there may exist more than one solutions having different values of other objectives, in problems having more than two objectives. In these problems, the payoff table method may not result in an accurate estimation of the nadir objective vector.

Let us consider the Pareto-optimal front of a hypothetical problem involving three objective functions shown in Figure 2. The problem has a bounded objective space lying inside the rectangular outer box marked with solid lines. The region below the triangular surface ABC is then removed from the box. Since all three objectives are minimized, the Pareto-optimal front is the triangular plane ABC. The minimum value of the first objective function is zero. As can be seen from the figure, there exist a number of solutions having a value zero for function f_1 and different combinations of f_2 and f_3 values. These solutions lie on the $f_1 = 0$ plane, but on the trapezoid CBB′F′C′C). In the payoff table, when three objectives are minimized one at a time, we may get objective vectors $\mathbf{f}^{(1)} = (0, 0, 1)^T$ (point C), $\mathbf{f}^{(2)} = (1, 0, 0)^T$ (point A), and $\mathbf{f}^{(3)} = (0, 1, 0)^T$ (point B) corresponding to minimizations of f_1, f_2, and f_3, respectively, and then the true nadir point $z^{\text{nad}} = (1, 1, 1)^T$ can be found. However, if vectors $\mathbf{f}^{(1)} = (0, 0.2, 0.8)^T$, $\mathbf{f}^{(2)} = (0.5, 0, 0.5)^T$ and $\mathbf{f}^{(3)} = (0.7, 0.3, 0)^T$ (marked with open circles) are found from corresponding minimizations of f_1, f_2, and f_3, respectively, a wrong estimate $\mathbf{z}' = (0.7, 0.3, 0.8)^T$ of the nadir point will be made. The figure shows how such a wrong nadir point represents only a portion (shown dark-shaded) of the Pareto-optimal front. Here we obtained an underestimation but the result may also be an overestimation of the true nadir point in some other problems. Thus, we need a more reliable method to estimate the nadir point.

3 Evolutionary Multi-Objective Approaches for Nadir Point Estimation

As has been discussed so far, the nadir point is associated with Pareto-optimal solutions and, thus, determining a set of Pareto-optimal solutions will facilitate the estimation of the nadir point. For the past decade or so, EMO algorithms have been gaining popularity because of their ability to find multiple, wide-spread, Pareto-optimal solutions simultaneously [11, 6]. Since they aim at finding a set of Pareto-optimal solutions, an EMO approach may be an ideal way to find the nadir objective vector. Let us now discuss several approaches for estimating the nadir point.

3.1 Naive Approach

In the so-called naive approach, first a well-distributed set of Pareto-optimal solutions can be attempted to find by an EMO, as was also suggested in [15]. Thereafter, an estimate of the nadir objective vector can be made by picking the worst values of each objective. This idea was implemented in [37] and applied to a couple of three and four objective optimization problems. However, this naive approach of first finding a representative set of Pareto-optimal solutions and then determining the nadir objective vector seems to possess some difficulties. In the context of the problem depicted in Figure 2, this means first finding a well-represented set of solutions on the plane ABC and then estimating the nadir point from them.

Recall that one of the main purposes of estimating the nadir objective vector is that along with the ideal point, it can be used to normalize different objective functions, so that an interactive multi-objective optimization algorithm can be used to find the most preferred Pareto-optimal solution. But by the naive approach, an EMO is already utilized to find a representative set of Pareto-optimal solutions. One may think that there is no apparent reason for constructing the nadir point for any further analysis. However, representing and analyzing the set of Pareto-optimal solutions is not trivial when we have more than two objectives in question. Furthermore, we can list several other difficulties related to the above-described simple approach. Recent studies have shown that EMO approaches using the domination principle possess a number of difficulties in solving problems having a large number of objectives [25, 18, 7]:

1. To properly represent a high-dimensional Pareto-optimal front requires an exponentially large number of points [11], which, among others, increases computational cost.

2. With a large number of conflicting objectives, a large proportion of points in a random initial population are non-dominated to each other. Since EMO algorithms emphasize *all* non-dominated solutions in a generation, a large portion of an EA population gets copied to the next generation, thereby allowing only a small number of new solutions to be included in a generation. This severely slows down the convergence of an EMO towards the true Pareto-optimal front.

3. EMO methodologies maintain a good diversity of non-dominated solutions by explicitly using a niche-preserving scheme which uses a diversity metric specifying how diverse the non-dominated solutions are. In a problem with many objectives, defining a computationally fast yet a good indicator of higher-dimensional distances among solutions becomes a difficult task. This aspect also makes the EMO approaches computationally expensive.

4. With a large number of objectives, visualization of a large-dimensional Pareto-optimal front gets difficult.

The above-mentioned shortcomings cause EMO approaches to be inadequate for finding the complete Pareto-optimal front in the first place [18]. Thus, for handling a large number of objectives, it may not be advantageous to use the naive approach in which an EMO is employed to first find a representative set of points on the entire Pareto-optimal front and then construct the nadir point from these points.

3.2 Multiple Bi-Objective Formulations

Szczepanski and Wierzbicki [37] have simulated the idea of solving multiple bi-objective optimization problems suggested in [21] using an EMO approach and constructing the nadir point by accumulating all bi-objective Pareto-optimal fronts together. In the context of the three-objective optimization problem described in Figure 2 for which the Pareto-optimal front is the plane ABC, minimization of the pair f_1-f_2 will correspond to one Pareto-optimal objective vector having a value of zero for both objectives. An easy way to visualize the objective space for the f_1-f_2 optimization problem is to project every point on the above three-dimensional objective space on the f_1-f_2 plane. The projected objective space lies on the first quadrant of the f_1-f_2 plane and the origin (the point (0,0) corresponding to (f_1, f_2)) is the only Pareto-optimal point to the above problem. However, this optimal objective vector ($f_1 = 0$ and $f_2 = 0$) corresponds to any value of the third objective function lying on the line CC$'$ (since the third objective was not considered in the above bi-objective optimization process). The authors of [37] have also suggested the use an objective-space *niching* technique to find a set of well-spread optimal solutions on the objective space. But since all objective vectors on the line CC$'$ correspond to an identical (f_1, f_2) value of (0,0), the objective-space niching will not have any motivation to find multiple solutions on the line CC$'$. Thus, to find multiple solutions on the line CC$'$ so that the point C can be captured by this bi-objective optimization task to make a correct estimate of the nadir point, an additional *variable-space niching* [16, 10] must also be used to get a well-spread set of solutions on the line CC$'$. This aspect was ignored in the original study [37], but it is important to note that in order to accurately estimate the nadir point, any arbitrary objective vector on the line CC$'$ will not be adequate, but the point C must be found. Similarly, the other two pair-wise minimizations, if performed with a variable-space niching, will give rise to sets of solutions on the lines AA$'$ and BB$'$. According to the procedure of [37], all these points (objective vectors) can then be put together, dominated solutions can be eliminated, and the nadir point can be estimated from the remaining non-dominated points. If only objective vectors A, B and C are found by respective pair-wise minimizations exactly, the above procedure will result in three non-dominated solutions A, B, and C, thereby making a correct estimate of the nadir point.

Although the idea seems interesting and theoretically sound, it requires $\binom{M}{2}$ bi-objective optimizations with both objective and variable-space niching methodologies to be performed. This may be a daunting task particularly for problems having more than three or four objectives. Moreover, the outcome of the procedure will depend on the chosen niching parameter on both objective and decision-space niching operators.

However, the idea of concentrating on a preferred region on the Pareto-optimal front, instead of finding the entire Pareto-optimal front, can be pushed further. Instead of finding bi-objective Pareto-optimal fronts by several pair-wise optimizations, an emphasis can be placed in an EMO approach to find only the critical points of the Pareto-optimal front. These points are non-dominated points which will be required to estimate the nadir point correctly. With this change in focus, the EMO approach can also be used to handle large-dimensional problems, particularly since the focus would be to only converge to the extreme points on the Pareto-optimal front, instead of aiming at maintaining diversity. For the three-objective minimization problem of Figure 2, the proposed EMO approach would then distribute its population members near the extreme points A, B, and C (instead of the entire Pareto-optimal front or non-optimal solutions), so that the nadir point can be estimated quickly. Our earlier study [15] suggested the following two approaches.

3.3 Worst-Crowded NSGA-II Approach

We implemented this approach on a particular EMO approach (NSGA-II [13]), but the concept can, in principle, be implemented on other state-of-the-art EMO approaches as well. Since the nadir point must be constructed from the worst objective values of Pareto-optimal solutions, it is intuitive to think of an idea in which population members having the worst objective values within a non-dominated front are emphasized. For this, we employed a modified *crowding distance scheme* in NSGA-II by emphasizing the worst objective values in every non-dominated front. We called this by the name 'Worst-Crowded NSGA-II Approach'.

In every generation, population members on every non-dominated front (having N_f members) are first sorted from minimum to maximum based on each objective (for minimization problems) and a rank equal to the position of the solution in the sorted list is assigned. In this way, a member i in a front gets a rank $R_i^{(m)}$ from the sorting in the m-th objective. The solution with the minimum function value in the m-th objective gets a rank value $R_i^{(m)} = 1$ and the solution with the maximum function value in the m-th objective gets a rank value $R_i^{(m)} = N_f$. Such a rank assignment continues for all M objectives. Thus, at the end of this assignment process, each solution in the front gets M ranks, one corresponding to each objective function. Thereafter, the crowding distance d_i to a solution i in the front is assigned as the maximum of all M ranks:

$$d_i = \max\left\{R_i^{(1)}, R_i^{(2)}, \ldots, R_i^{(M)}\right\}. \tag{2}$$

In this way, the solution with the maximum objective value of any objective gets the best crowding distance. The NSGA-II approach emphasizes a solution if it lies on a better non-dominated front and for solutions of the same non-dominated front it emphasizes a solution with a higher crowding distance value. Thus, solutions of the final non-dominated front which could not be accepted entirely by NSGA-II's selection operator are chosen based on their crowding distance value. This dual task of selecting non-dominated solutions and solutions with worst objective values should, in principle, lead to a proper estimation of the nadir point in most problems.

However, we realize that an emphasis on the worst non-dominated points alone may have at least two difficulties in certain problems. First, since the focus is to find only a few solutions (instead of a complete front), the population may lose its diversity early on during the search process, thereby slowing down the progress towards the true worst points. Moreover, if, for some reason, the convergence is a premature event to wrong solutions, the lack of diversity among population members will make it even harder for the EMO to recover and find the necessary worst solutions to construct the true nadir point.

The second difficulty of the worst-crowded NSGA-II approach may appear in certain problems, in which an identification of critical points alone from the Pareto-optimal front is not enough. Some spurious non-Pareto-optimal points can remain non-dominated with the critical points in a population and provide a wrong estimate of the nadir point. Let us next discuss this important issue with an example problem. Consider a three-objective minimization problem shown in Figure 3, where the surface ABCD represents the Pareto-optimal front. The true nadir point is at $\mathbf{z}^{\rm nad} = (1, 1, 1)^T$. By using the worst-crowded NSGA-II, we expect to find three individual critical points: B=$(1, 0, 0.4)^T$ (for f_1), D=$(0, 1, 0.4)^T$ (for f_2) and C=$(0, 0, 1)^T$ (for f_3). Note that there is no motivation for the worst-crowded NSGA-II to find and maintain point A=$(0.9, 0.9, 0.1)^T$ in the population, as this point does not correspond to the worst value of any objective in the set of Pareto-optimal solutions. With the three points (B, C, and D) in a

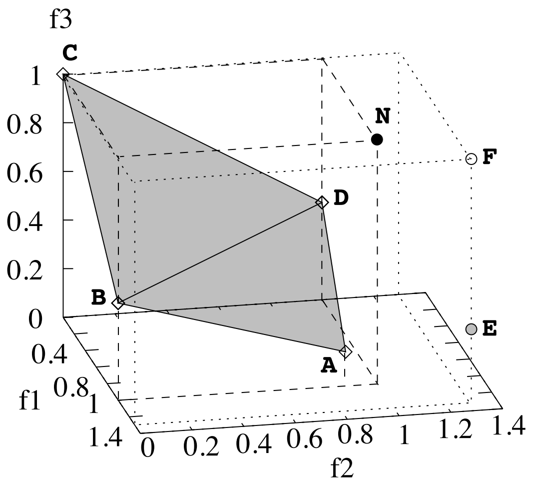

Figure 3: A problem which may cause difficulty to the worst-crowded approach.

population, a non-Pareto-optimal point E (with an objective vector $(1.3, 1.3, 0.3)^T$), if found by EA operators, will become non-dominated to points B, C, and D, and will continue to exist in the population. Thereafter, the worst-crowded NSGA-II will emphasize points C and E as extreme points and the reconstructed nadir point will become F=$(1.3, 1.3, 1.0)^T$, which is a wrong estimation. This difficulty could have been avoided, if the point A was included in the population.

A little thought will reveal that the point A is a Pareto-optimal solution, but corresponds to the best value of f_3. If the point A is present in the population, it will dominate the point E and would not allow points like E to be present in the non-dominated front. Interestingly, this situation does not occur in bi-objective optimization problems. To avoid a wrong estimation of the nadir point due to the above difficulty, ideally, an emphasis on maintaining *all* Pareto-optimal solutions in the population must be made. But, since this is not practically viable for a large number of objectives, we suggest another approximate approach which deals with the above-mentioned difficulties better than the worst-crowded approach.

3.4 Extremized-Crowded NSGA-II Approach

In the extremized-crowded NSGA-II approach, in addition to emphasizing the worst solution corresponding to each objective, we also emphasized the best solution corresponding to every objective. We refer to the individual best and worst Pareto-optimal solutions as 'extreme' solutions here. In the extremized crowded NSGA-II approach, solutions on a particular non-dominated front are first sorted from minimum (with rank $R_i^{(m)} = 1$) to maximum (with rank $= N_f$) based on each objective. A solution closer to either extreme objective values (minimum or maximum objective values) gets a higher rank compared to that of an intermediate solution. Thus, the rank of solution i for the m-th objective $R_i^{(m)}$ is reassigned as $\max\{R_i^{(m)}, N_f - R_i^{(m)} + 1\}$. Two extreme solutions for every objective get a rank equal to N_f (number of solutions in the non-dominated front), the solutions next to these extreme solutions get a rank $(N_f - 1)$, and so on. Figure 4 shows this rank-assignment procedure. After a rank is assigned to a solution by each

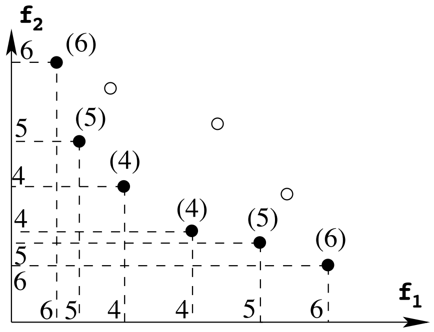

Figure 4: Crowding distance computation procedure in extremized-crowded NSGA-II approach.

objective, the maximum value of the assigned ranks is declared as the crowding distance, as in (2). The final crowding distance values are shown within brackets in Figure 4.

For a problem having a one-dimensional Pareto-optimal front (such as, in a bi-objective problem), the above crowding distance assignment is similar to the worst crowding distance assignment scheme (as the minimum-rank solution of one objective is also the maximum-rank solution of at least one other objective). However, for problems having a higher-dimensional Pareto-optimal hyper-surface, the effect of extremized crowding is different from that of the worst-crowded approach. In the three-objective problem shown in Figure 3, the extremized-crowded approach will not only emphasize the extreme points A, B, C and D, but also solutions on edges CD and BC (having the smallest f_1 and f_2 values, respectively) and solutions near them. This approach has two advantages: (i) a diversity of solutions in the population may now allow genetic operators (recombination and mutation) to find better solutions and not cause a premature convergence and (ii) the presence of these extreme solutions will reduce the chance of having spurious non-Pareto-optimal solutions (like point E in Figure 3) to remain in the non-dominated front, thereby enabling a more accurate computation of the nadir point. Moreover, since the intermediate portion of the Pareto-optimal front is not targeted in this approach, finding the extreme solutions should be quicker than the original NSGA-II, especially for problems having a large number of objectives and involving computationally expensive function evaluation schemes.

4 Nadir Point Estimation Procedure

An accurate estimation of the nadir point depends on how accurately the critical points can be found. For solving multi-objective optimization problems, the NSGA-II approach (and for this matter any other EMO approach) is usually found to come close to the Pareto-optimal front quickly and then observed to take many iterations to reach the exact front [28]. To accurately determine the Pareto-optimal front, NSGA-II solutions can be improved by using a local search approach [11, 36]. For estimating the nadir point accurately, we propose to employ an EMO-cum-local-search approach, in which the solutions obtained by the modified NSGA-II approaches will be attempted to be improved by using a local-search procedure.

4.1 Bilevel Local Search Approach

Recall that due to the focus of modified NSGA-II approaches towards individual objective-wise worst or extreme solutions, the algorithms are likely to find solutions close to the critical point for each objective. The task of the local search would then be to take each of these solutions to the corresponding critical point as close as possible. Particularly we would like to have following three goals of our local search procedure. First, the approach must be generic, so that it, for example, be applicable to convex and non-convex problems. Second, the approach must guarantee convergence to a Pareto-optimal point, no matter which solutions were found by the modified NSGA-II approach. Third, the approach must find that particular Pareto-optimal solution which corresponds to the worst value of the underlying objective. It is clear that the above task of the local search procedure involves two optimization problems (to ensure the second and the third properties, respectively). Thus, the local search approach must be different from the usual local search methods employed in the EMO algorithm. Somehow, both optimization tasks must be combined together in a generic manner so that the local search can be applied to different types of problems and is able to accurately find the critical points leading to the nadir point.

The first two properties can be achieved by using a well-known MCDM approach, called the augmented achievement scalarizing function approach [38]. In this approach, a reference point $\mathbf{z}$ is first chosen. By using a weight vector $\mathbf{w}$ (used for scaling), the following minimization problem is then solved:

$$\begin{array}{rl} \text{minimize} & \max_{j=1}^{M} w_j(f_j(\mathbf{x})-z_j)+\rho\sum_{j=1}^{M} w_j(f_j(\mathbf{x})-z_j), \\ \text{subject to} & \mathbf{x}\in\mathcal{S}, \end{array} \tag{3}$$

where $\mathcal{S}$ is the original set of feasible solutions. The right-most augmented term in the objective function is added so that a weak Pareto-optimal solution is not found. For this purpose, a small value of ρ (e.g., 10^{-4} or smaller) is used. The above optimization task involves a non-differentiable objective function (due to the max-term in the objective function), but if the original problem is differentiable, a suitable transformation of the problem can be made by introducing an additional slack variable x_{n+1} to make an equivalent differentiable problem [30], as follows:

$$\begin{array}{rl} \text{minimize} & x_{n+1}+\rho\sum_{j=1}^{M} w_j(f_j(\mathbf{x})-z_j), \\ \text{subject to} & x_{n+1}\geq w_j(f_j(\mathbf{x})-z_j), \quad j=1,2,\ldots,M, \\ & \mathbf{x}\in\mathcal{S}. \end{array} \tag{4}$$

If the single-objective optimization algorithm used to solve the above problem is able to find the true optimum, the optimal solution is guaranteed to be a Pareto-optimal solution [30]. In other words, achievement scalarizing functions project the reference point on the Pareto-optimal front. Moreover, the above approach is applicable for both convex and non-convex problems. Figure 5 illustrates the idea. For the reference point C, the optimal solution of the above problem is D, which is a Pareto-optimal point. The direction marked by the arrow depends on the chosen weight vector $\mathbf{w}$. Irrespective of whether the reference point is feasible or not, the approach always finds a Pareto-optimal point dictated by the chosen weight vector and the reference point. The effect of the augmented term (with the term involving ρ) is shown by plotting a sketch of the iso-preference contour lines. More information about the role of weights is given, for example, in [29].

However, our goal for the local search is not to find any arbitrary Pareto-optimal solution, but the critical point corresponding to the underlying objective (like the point P for objective f_2). Unfortunately it is not obvious which reference point and weight vector one must choose to arrive at a critical point. For this purpose, we construct another optimization problem to find a combination of a reference point and a weight vector which will result in the critical point for an objective. This requires a nested *bilevel* approach in which the upper-level optimization considers a combination of a reference point and a weight vector $(\mathbf{z},\mathbf{w})$ as decision variables. Each combination $(\mathbf{z},\mathbf{w})$ is then evaluated by finding a Pareto-optimal solution corresponding to a lower-level optimization problem constructed using an augmented achievement scalarizing function given in (3) or (4) with $\mathbf{z}$ and $\mathbf{w}$ as the reference point and the weight vector, respectively. In the lower-level optimization, problem variables ($\mathbf{x}$) are the decision variables. As discussed above, the resulting optimal solution of the lower-level optimization is always a Pareto-optimal solution (having a objective vector $\mathbf{f}^*$). Since our goal in the local search approach is to reach the critical point corresponding to a particular objective (say j-th objective), a solution ($\mathbf{z}$,$\mathbf{w}$) for the upper-level optimization task can be evaluated by checking the j-th objective value (f_j^*) of the obtained Pareto-optimal solution.

Figure 5 further explains this bilevel approach. Consider points A and B which are found by one of the modified NSGA-II procedures as worst non-dominated solutions for f_2 and f_1, respectively. The

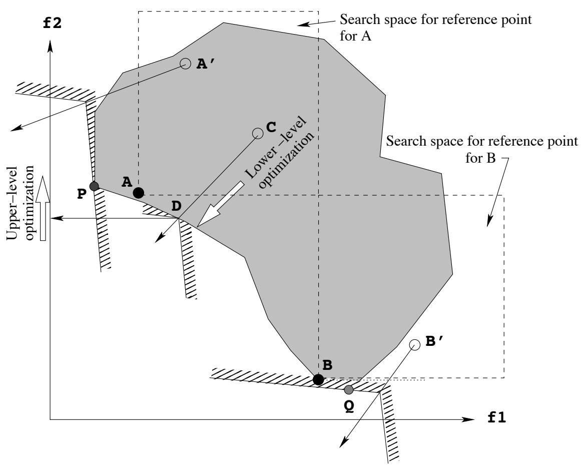

Figure 5: The bilevel local search procedure. A and B are critical points obtained by EMO. The task of local search is to find point P from A and point Q from B for an accurate estimate of the nadir point.

goal of using the local search approach is to reach the corresponding critical points (P and Q, respectively) from each of these points. Consider point A, which is found to be the worst in objective f_2 among all modified NSGA-II solutions. The search region for the reference point $\mathbf{z}$ in the upper-level optimization is shown by the dashed box for which A is the lower-left corner point. Each component of the weight vector ($\mathbf{w}$) is restricted within a non-negative range of values ($[0.001, 1.000]$ is chosen for this study). For the reference point $\mathbf{z}$, say C, and weight vector $\mathbf{w}$ (directions indicating improvement of achievement scalarizing function), the solution to the lower-level optimization problem (problem (3) or (4)) is the decision variable vector $\mathbf{x}$ corresponding to solution D. Thus, for the reference point C and the chosen weight vector, the corresponding function value of the upper-level optimization problem is the objective value f_2 of D. Since this objective value always corresponds to a Pareto-optimal solution and the upper-level optimization attempts to maximize this objective value, intuitively, the upper-level optimization will eventually result in finding the point P (the critical point for f_2). It is interesting to note that there may exist many combinations of ($\mathbf{z}$,$\mathbf{w}$) (for example, with reference point A′ and weight vector shown in the figure) which will also result in the same point P and for our purpose any one of such solutions would be adequate to accurately estimate the nadir point. Similarly, for the modified NSGA-II solution B (worst for objective f_1), the point Q will be the outcome of the above bilevel optimization approach, resulting from a possible combination of the reference point B′ and the weight vector shown in the figure. Because we solve the single-objective lower-level problems ((3) or (4)) with an appropriate local optimization algorithm and the task of the upper-level search is also restricted in a local neighborhood, we refer to this bilevel search as a local search operation.

Now we are ready to outline the overall nadir point estimation procedure in a step-by-step format:

Step 1: Supply or compute ideal and worst objective vectors by minimizing and maximizing each objective function independently within the set of feasible solutions.

Step 2: Apply the worst-crowded or the extremized-crowded NSGA-II approach to find a set of non-dominated points. Iterations are continued till a termination criterion (described in the next subsection), which uses ideal and worst objective vectors computed in Step 1, is met. Say, P non-dominated extreme points (variable vector $\mathbf{x}_{\mathrm{EA}}^{(i)}$ with objective vector $\mathbf{f}_{\mathrm{EA}}^{(i)}$ for $i = 1, 2, \ldots, P$) are found in this step. Identify the minimum and maximum objective vectors ($\mathbf{f}^{\min}$ and $\mathbf{f}^{\max}$) from the P obtained extreme solutions. For the j-th objective, they are computed as follows:

$$f_j^{\min} = \min_{i=1}^{P} f_{j\ \mathrm{EA}}^{(i)}, \tag{5}$$

$$f_j^{\max} = \max_{i=1}^{P} f_{j\ \mathrm{EA}}^{(i)}. \tag{6}$$

Step 3: Apply the bilevel local search approach for each objective j ($\in \{1, \ldots, M\}$), one at a time. First, identify the objective-wise worst solution (solution $\mathbf{x}_{\mathrm{EA}}^{(j)}$ for which the j-th objective has the worst value in P) and then find the corresponding optimal solution $\mathbf{y}^{*(j)}$ in the variable space by using the bilevel local search procedure, as follows. The upper-level optimization uses a combination of a reference point and a weight vector ($\mathbf{z}, \mathbf{w}$) as the decision variables and maximizes the j-th objective value of the Pareto-optimal solution obtained by the lower-level optimization task (described a little later):

$$\begin{array}{rl} \text{maximize}_{(\mathbf{z},\mathbf{w})} & f_j^*(\mathbf{z},\mathbf{w}), \\ \text{subject to} & 0.001 \leq w_j \leq 1, \quad j = 1,2,\ldots,M, \\ & z_i \geq f_i^{(j)}{}_{\text{EA}}, \quad i = 1,2,\ldots,M, \\ & z_i \leq f_i^{(j)}{}_{\text{EA}} + (f_i^{\max} - f_i^{\min}), \\ & \quad i = 1,2,\ldots,M. \end{array} \tag{7}$$

The term $f_j^*(\mathbf{z},\mathbf{w})$ is the value of the j-th objective function at the optimal solution to the following lower-level optimization problem:

$$\begin{array}{rl} \text{minimize}_{(\mathbf{y})} & \max_{i=1}^{M} w_i \left(\frac{f_i(\mathbf{y})-z_i}{f_i^{\max}-f_i^{\min}}\right) \\ & +\rho \sum_{k=1}^{M} w_k \left(\frac{f_k(\mathbf{y})-z_k}{f_k^{\max}-f_k^{\min}}\right), \\ \text{subject to} & \mathbf{y} \in \mathcal{S}. \end{array} \tag{8}$$

This problem is identical to that in equation (3), except that individual objective terms are normalized for a better property of the augmented term. In this lower-level optimization problem, the search is performed on the original decision variable space. The solution $\mathbf{y}*^{(j)}$ to this lower-level optimization problem determines the optimal objective vector $\mathbf{f}(\mathbf{y}*^{(j)})$ from which we extract the j-th component and use it in the upper-level optimization problem. Thus, for every reference point $\mathbf{z}$ and weight vector $\mathbf{w}$, considered in the upper-level optimization task, the corresponding optimal augmented achievement scalarizing function is found in the lower-level loop. The upper-level optimization is initialized with the NSGA-II solution $\mathbf{z}^{(0)} = \mathbf{f}(\mathbf{x}_{\text{EA}}^{(j)})$ and $w_i^{(0)} = 1/M$. The lower-level optimization is initialized with the NSGA-II solution $\mathbf{y}^{(0)} = \mathbf{x}_{\text{EA}}^{(j)}$. The local search can be terminated based on standard single-objective convergence measures, such as Karush-Kuhn-Tucker (KKT) condition satisfaction through a prescribed limit or a small difference in variable vectors between successive iterations.

Step 4: Finally, construct the nadir point from the worst objective values of the all Pareto-optimal solutions obtained after the local search procedure.

The use of a bilevel local search approach can be computationally expensive, if the starting solution to the local search is far away from the critical point. For this reason, the proposed local search procedure may not be computationally viable if started from a random initial point. However, the use of a modified NSGA-II approach to first find a near critical point and then to employ the proposed local search to accurately locate the critical point seems like a viable approach. To demonstrate the computational viability of using the proposed local search approach within our nadir point estimation procedure, we shall present a break-up of function evaluations needed by both NSGA-II and local search procedures later.

Before we leave this subsection, we discuss one further issue. It is mentioned above that the use of the augmentation term in the achievement scalarizing problem formulation allows us not to converge to a weak Pareto-optimal solution by the local search approach. But, in certain problems, the approach may only find a critical *proper* Pareto-optimal solution [30] depending on the value of the parameter ρ. For this reason, we actually get an estimate of the ranges of objective function values in a properly Pareto-optimal set and not in a Pareto-optimal set. We can control the trade-offs in the properly Pareto-optimal set by choosing an appropriately small ρ value. For further details, see, for example, [30]. In certain problems having a small trade-off near the critical points, a proper Pareto-optimal point can be away from the true critical point. If this is not desired, it is possible to solve a lexicographic achievement scalarizing function [30, 34] instead of the augmented one suggested in Step 3.

4.2 Termination Criterion for Modified NSGA-II

Typically, a NSGA-II run is terminated when a prespecified number of generations is elapsed. Here, we suggest a performance based termination criterion which causes a NSGA-II run to stop when the performance reaches a desirable level. The performance metric depends on a measure stating how close the estimated nadir point is to the true nadir point. However, for applying the proposed NSGA-II approaches to an arbitrary problem (for which the true Pareto-optimal front, hence the true nadir point, is not known a priori), we need a different concept. Using the ideal point ($\mathbf{z}^*$), the worst objective vector ($\mathbf{z}^w$), and the estimated nadir point $\mathbf{z}^{\text{est}}$ at any generation of NSGA-II, we can define a *normalized distance* (*ND*) metric as follows and track the convergence property of this metric to determine the termination of our NSGA-II approach:

$$ND = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\frac{z_i^{\text{est}} - z_i^*}{z_i^w - z_i^*}\right)^2}. \tag{9}$$

If in a problem, the worst objective vector $\mathbf{z}^w$ (refer to Figure 1) is the same as the nadir point, the normalized distance metric value must converge to one. Since the exact final value of this metric for finding the true nadir point is not known a priori on an arbitrary problem, we record the change in *ND* for the past τ generations. Say, $ND_{\max}$, $ND_{\min}$, and ND_{avg}, are the maximum, minimum, and average *ND* values for the past consecutive τ generations. If the change $(ND_{\max} - ND_{\min})/ND_{\text{avg}}$ is smaller than a threshold Δ, the proposed NSGA-II approach is terminated and the current non-dominated extreme solutions are sent to the next step for performing the local search.

However, in the case of solving some academic test problems, the location of the nadir objective vector is known and a simple *error* metric (E) between the estimated and the known nadir objective vectors can be used for stopping a NSGA-II run:

$$E = \sqrt{\sum_{i=1}^{M} \left(\frac{z_i^{\text{nad}} - z_i^{\text{est}}}{z_i^{\text{nad}} - z_i^{*}} \right)^2}. \tag{10}$$

To make the approach pragmatic, in this paper, we terminate a NSGA-II run when the error metric E becomes smaller than a predefined threshold (η), whenever the true nadir point is known.

5 Results on Benchmark Problems

We are now ready to describe the results of numerical tests obtained using the proposed hybrid nadir point estimation procedure. We have chosen problems having three to 20 objectives in this study. In this section, we use benchmark problems where the entire description of the objective space and the Pareto-optimal front is known. We have chosen these problems to test the working of our procedure. Thus, in these problems, we do not perform Step 1 explicitly. Moreover, if Step 2 of the nadir point estimation procedure successfully finds the nadir point (using the error metric ($E \leq \eta$) for determining termination of a run), we do not employ Step 3 (local search). The complete hybrid procedure will be tested in its totality in the next section.

In all runs here, we compare three different approaches:

1. Naive NSGA-II approach in which first we find a set of non-dominated solutions using the original NSGA-II and then estimate the nadir point from the obtained solutions.
2. NSGA-II with the worst-crowded approach, and
3. NSGA-II with the extremized-crowded approach.

To make a fair comparison, parameters in all three cases are kept fixed for all problems. We use the SBX recombination operator [12] with a probability of 0.9 and distribution index of $\eta_c = 10$. The polynomial mutation operator [11] is used with a probability of $1/n$ (n is the number of variables) and a distribution index of $\eta_m = 20$. The population size is set according to the problem and is mentioned in respective subsections. Each algorithm is run 11 times (odd number of runs are used to facilitate the recording of the median performance of an algorithm), each time starting from a different random initial population. However all proposed procedures are started with an identical set of initial populations to be fair. The number of generations required to satisfy the termination criterion ($E \leq \eta$) is noted for each run and the corresponding best, median and worst number of generations are presented for a comparison. For all test problems, $\eta = 0.01$ is used.

5.1 Three and More Objectives

To test Step 2 of the nadir point estimation procedure on three and more objectives, we choose three DTLZ test problems [19] which have different characteristics. These problems are designed in a manner so that they can be extended to any number of objectives. The first problem, DTLZ1, is constructed to have a linear Pareto-optimal front. The true nadir objective vector is $\mathbf{z}^{\text{nad}} = (0.5, \ldots, 0.5)^T$ and the ideal objective vector is $\mathbf{z}^* = (0, \ldots, 0)^T$. The Pareto-optimal front of the second test problem, DTLZ2, is a quadrant of a unit sphere centered at the origin of the objective space. The nadir objective vector is $\mathbf{z}^{\text{nad}} = (1, \ldots, 1)^T$ and the ideal objective vector is $\mathbf{z}^* = (0, \ldots, 0)^T$. The third test problem, DTLZ5, is somewhat modified from the original DTLZ5 and has a one-dimensional Pareto-optimal curve in the M-dimensional space [18]. The ideal objective vector is $\mathbf{z}^* = (0, \ldots, 0)^T$ and the nadir objective vector is $\mathbf{z}^{\text{nad}} = \Big((\frac{1}{\sqrt{2}})^{M-2}, (\frac{1}{\sqrt{2}})^{M-2}, (\frac{1}{\sqrt{2}})^{M-3}, (\frac{1}{\sqrt{2}})^{M-4}, \ldots, (\frac{1}{\sqrt{2}})^{0} \Big)^T$.

5.1.1 Three-Objective DTLZ Problems

All three approaches are run with 100 population members for problems DTLZ1, DTLZ2 and DTLZ5 involving three objectives. Table 1 shows the numbers of generations needed to find a solution close (within an error metric value of $\eta = 0.01$ or smaller) to the true nadir point. It can be observed that the worst-crowded NSGA-II and the extremized crowded NSGA-II perform in a more or less similar way when compared to each other and are somewhat better than the naive NSGA-II approach. In the DTLZ5 problem, despite having three objectives, the Pareto-optimal front is one-dimensional [19]. Thus, the naive NSGA-II approach performs almost as well as the proposed modified NSGA-II approaches.

To compare the working principles of the two modified NSGA-II approaches and the naive NSGA-

Table 1: Comparative results for DTLZ problems with three objectives.

Test problem	Pop. size	Number of generations								
		NSGA-II			Worst-crowd. NSGA-II			Extr.-crowd. NSGA-II		
		Best	Median	Worst	Best	Median	Worst	Best	Median	Worst
DTLZ1	100	223	366	610	171	282	345	188	265	457
DTLZ2	100	75	111	151	38	47	54	41	49	55
DTLZ5	100	63	80	104	59	74	86	62	73	88

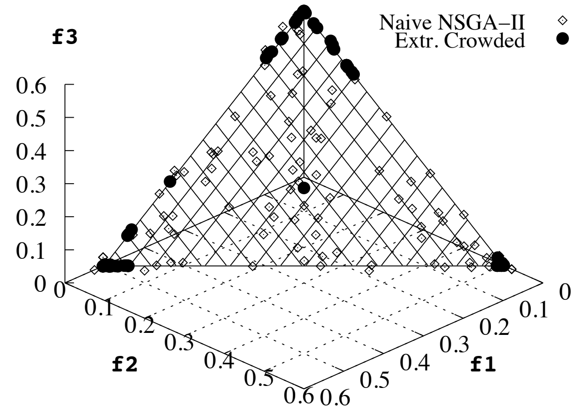

Figure 6: Populations obtained using extremized-crowded and naive NSGA-II for DTLZ1. Extremized crowded NSGA-II finds the objective-wise extreme points, whereas naive NSGA-II approach finds a distributed set of points.

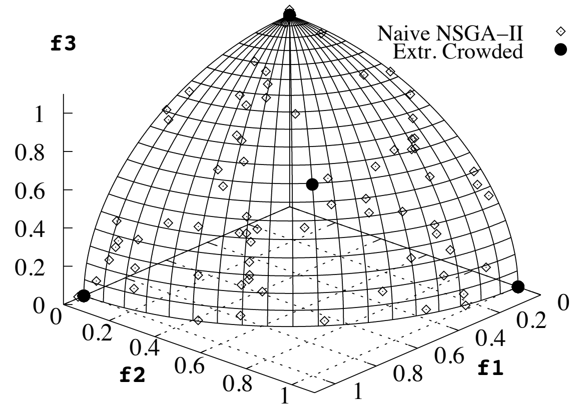

Figure 7: Populations obtained using extremized-crowded and naive NSGA-II for DTLZ2. Extremized crowded NSGA-II finds objective-wise extreme points.

II approach, we show the final populations for the extremized-crowded NSGA-II and the naive NSGA-II for DTLZ1 and DTLZ2 in Figures 6 and 7, respectively. Similar results are also found for the worst-crowded NSGA-II approach, but are not shown here for brevity. It is clear that the extremized-crowded NSGA-II concentrates its population members near the extreme regions of the Pareto-optimal front, so that a quicker estimation of the nadir point is possible to achieve. However, in the case of the naive NSGA-II approach, a distributed set of Pareto-optimal solutions is first found using the original NSGA-II (as shown in the figure) and the nadir point is constructed from these points. Since the intermediate points do not help in constructing the nadir objective vector, the naive NSGA-II approach is expected to be computationally inefficient and also inaccurate, particularly for problems having a large number of objectives. There is not much of a difference in the performance of the original NSGA-II and modified NSGA-IIs for DTLZ5 problem due to two-dimensional nature of the Pareto-optimal front. Hence, we do not show the corresponding figure here.

To investigate if the error metric (E) deteriorates with generations, we continue to run the two modified NSGA-II procedures till 1,000 generations. For the DTLZ1 problem, the worst-crowded approach settles on an E value in the range [0.000200, 0.000283] for 11 independent runs and the extremized-crowded approach in the range [0.000199, 0.000283]. For DTLZ2, both approaches settle to $E = 0.000173$ and for DTLZ5, worst-crowded and extremized-crowded NSGA-IIs settle in the range [0.000211, 0.000768] and [0.000211, 0.000592], respectively. Since a threshold of $E \leq 0.01$ was used for termination in obtaining results in Table 1, respective NSGA-IIs terminated at a generation smaller than 1,000. However, these results show that there is no significant change in the nadir point estimation with the extra computations and the proposed procedure has a convergent property (which will also be demonstrated on higher objective problems through convergence metrics of this study in Figures 8 to 10, 13, 15 and 17).

5.1.2 Five-Objective DTLZ Problems

Next, we study the performance of all three NSGA-II approaches on DTLZ problems involving five objectives. In Table 2, we collect information about the results as in the previous subsection. It is now quite

Table 2: Comparative results for five and ten-objective DTLZ problems.

Test problem	Pop. size	Number of generations								
		NSGA-II			Worst-crowd. NSGA-II			Extr.-crowded NSGA-II		
		Best	Median	Worst	Best	Median	Worst	Best	Median	Worst
		Five-objective DTLZ problems								
DTLZ1	100	2,342	3,136	3,714	611	790	1,027	353	584	1,071
DTLZ2	100	650	2,142	5,937	139	166	185	94	114	142
DTLZ5	100	52	66	77	51	66	76	49	61	73
		Ten-objective DTLZ problems								
DTLZ1	200	17,581	21,484	33,977	1,403	1,760	2,540	1,199	1,371	1,790
DTLZ2	200	–	–	–	520	823	1,456	388	464	640
DTLZ5	200	45	53	60	43	53	57	45	51	64

evident from Table 2 that the modifications proposed to the NSGA-II approach perform much better than the naive NSGA-II approach. For example, for the DTLZ1 problem, the best NSGA-II run takes 2,342 generations to estimate the nadir point, whereas the extremized-crowded NSGA-II requires only 353 generations and the worst-crowded NSGA-II 611 generations. In the case of the DTLZ2 problem, the trend is similar. The median generation counts of the modified NSGA-II approaches for 11 independent runs are also much better than those of the naive NSGA-II approach.

The difference between the worst-crowded and extremized-crowded NSGA-II approaches is also clear from the table. For a problem having a large number of objectives, the extremized-crowded NSGA-II emphasizes both best and worst extreme solutions for each objective maintaining an adequate diversity among the population members. The genetic operators are able to exploit a relatively diversified population and make a faster progress towards the extreme Pareto-optimal solutions needed to estimate the nadir point correctly. However, on the DTLZ5 problem, the performance of all three approaches is similar due to the one-dimensional nature of the Pareto-optimal front. Figure 8 shows the convergence of the error metric value for the best runs of the three algorithms on DTLZ2. The figure demonstrates the convergent property of the proposed algorithm. The superiority of the extremized-crowded NSGA-II approach is clear from the figure. Similar results are also observed for DTLZ1. These results imply that for a problem having more than three objectives, an emphasis on the extreme Pareto-optimal solutions (instead of all Pareto-optimal solutions) is a faster approach for locating the nadir point.

So far, we have demonstrated the ability of the nadir point estimation procedure in converging close to the nadir point by tracking the error metric value which requires the knowledge of the true nadir point. It is clear that this metric cannot be used in an arbitrary problem. We have suggested a normalized distance metric (equation (9)) for this purpose. To demonstrate how the normalized distance metric can be used as a termination criterion, we record this metric value at every generation for both extremized crowded NSGA-II and the naive NSGA-II runs and plot them in Figure 9 for DTLZ2. Similar trends were observed for the worst-crowded NSGA-II and also for test problem DTLZ1, but for brevity these results are not shown here. To show the variation of the metric value over different initial populations, the region between the best and the worst normalized distance metric values is shaded and the median value is shown with a line. Recall that the normalized distance metric requires the information of the worst objective vector ($\mathbf{z}^w$). For the DTLZ2 problem, the worst objective vector is found to be $z_i^w = 3.25$ for $i = 1, \ldots, 5$. Figure 9 shows that the normalized distance metric (*ND*) value converges to around 0.286, which is identical to that computed by substituting the estimated nadir objective vector with the true nadir objective vector in equation (9). Thus, we can conclude that the convergence of the extremized-crowded NSGA-II is on the true nadir point. Despite the large variability in normalized distance value in different runs, all 11 runs of the extremized-crowded NSGA-II converge to the critical points at around 100 generations, indicating the robustness of the procedure. Similarity of this convergence pattern (at generation 100) with that in Figure 8 indicates that the normalized distance metric signifies convergence to the nadir point and can be used in arbitrary problems. The rate of convergence is also interesting to note from Figure 9. The extremized-crowded NSGA-II is able to find the nadir point much quicker (almost an order of magnitude faster) than the naive NSGA-II approach.

5.1.3 Ten-Objective DTLZ Problems

Next, we consider the three DTLZ problems for 10 objectives. Due to the increase in the dimension-

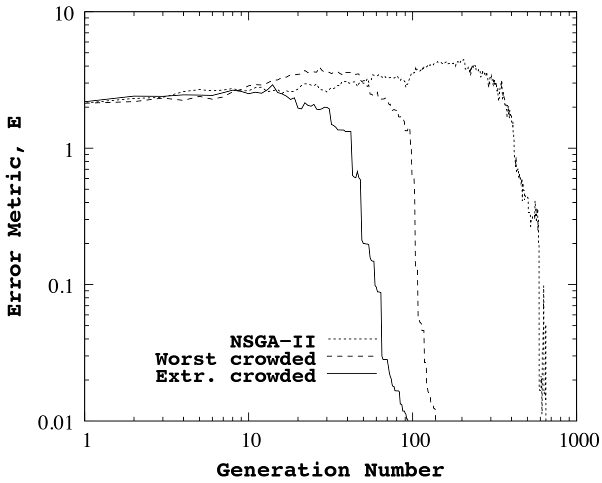

Figure 8: The error metric for best of 11 runs on five-objective DTLZ2. Extremized crowded NSGA-II is about an order of magnitude better than the naive NSGA-II approach.

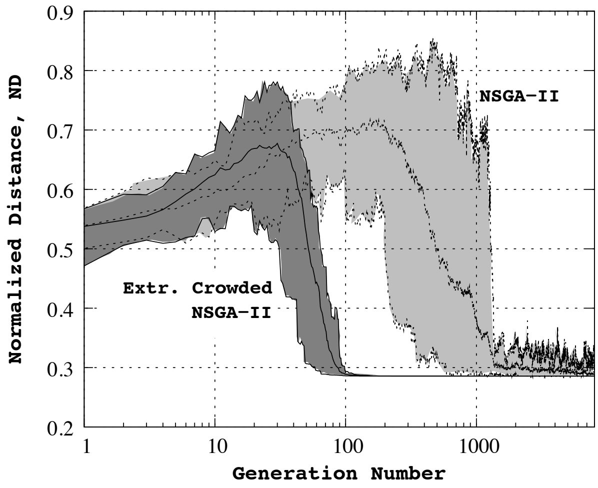

Figure 9: Variation of normalized distance metric in 11 runs for two methods on five-objective DTLZ2. Extremized crowded NSGA-II is about an order of magnitude better than the naive NSGA-II approach.

ality of the objective space, we double the population size for these problems. Table 2 presents the numbers of generations required to find a point close (within $\eta = 0.01$) to the nadir point by the three approaches for the DTLZ problems with ten objectives. It is clear that the extremized-crowded NSGA-II approach performs an order of magnitude better than the naive NSGA-II approach and is also better than the worst crowded NSGA-II approach. Both the DTLZ1 and DTLZ2 problems have 10-dimensional Pareto-optimal fronts and the extremized-crowded NSGA-II makes a good balance of maintaining diversity and emphasizing extreme Pareto-optimal solutions so that the nadir point estimation is quick. In the case of the DTLZ2 problem with ten objectives, the naive NSGA-II could not find the nadir objective vector even after 50,000 generations (and achieved an error metric value of 5.936). Figure 10 shows a typical convergence pattern of the extremized-crowded NSGA-II and the naive NSGA-II approaches on the 10-objective DTLZ1 problem. The figure demonstrates that for a large number of generations the estimated nadir point is away from the true nadir point, but after some generations (around 1,000 in this problem) the estimated nadir point comes quickly near the true nadir point. To understand the dynamics of the movement of the population in the best performed approach (the extremized-crowded NSGA-II) with the generation counter, we count the number of solutions in the population which dominate the true nadir point and plot this quantity in Figure 10. Points which dominate the nadir point lie in the region between the Pareto-optimal front and the nadir point. Thus, a task of finding these points is important to-

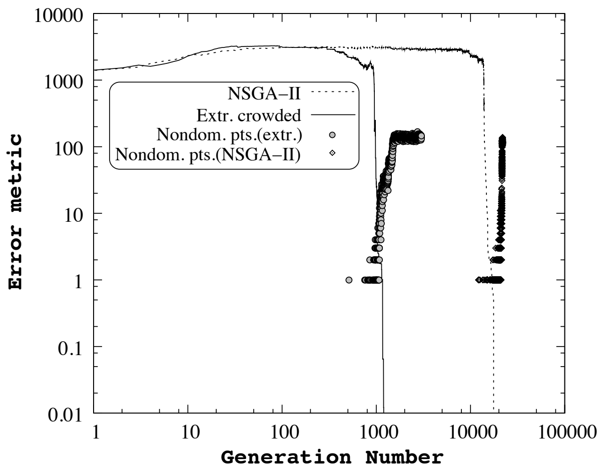

Figure 10: Performance of two methods on 10-objective DTLZ1. Extremized crowded NSGA-II is about an order of magnitude better than the naive NSGA-II approach. Convergence becomes faster after a solution dominating the nadir point is discovered.

wards reaching the critical points and therefore in estimating the nadir point. It is extremely unlikely to create such important points at random. But an optimization algorithm, starting with random solutions, must work towards finding such important points first before converging to the Pareto-optimal front. In DTLZ1, it is seen that the first point dominating the true nadir point appears in the population at around 750 generations with the extremized-crowded approach, whereas the naive NSGA-II needed about

10,000 generations. Thereafter, when an adequate number of such solutions start appearing in the population, the population very quickly converges near the critical points for correctly estimating the nadir point.

5.2 Scale-up Performance

Let us next investigate the overall function evaluations required to get near the true nadir point on DTLZ1 and DTLZ2 test problems having three to 20 objectives. As before, we use the stopping criterion $E \leq 0.01$. Here, we investigate the scale-up performance of the extremized crowded NSGA-II alone and compare it against that of the naive NSGA-II approach. Since the worst-crowded NSGA-II did not perform well on 10-objective DTLZ problems compared to the extremized-crowded NSGA-II approach, we do not apply it here.

Figure 11 plots the best, median, and worst of 11 runs of the extremized-crowded NSGA-II and the naive NSGA-II on DTLZ1. First of all, the figure clearly shows that the naive NSGA-II is unable to scale up to 15 or 20 objectives. In the case of 15-objective DTLZ1, the naive NSGA-II's performance is more than two orders of magnitude worse than that of the extremized-crowded NSGA-II. For this problem, the naive NSGA-II with more than 200 million function evaluations obtained a front having a poor error metric value of 12.871. Due to the poor performance of the naive NSGA-II approach on the 15-objective problem, we did not apply it to the 20-objective DTLZ1 problem.

Figure 12 shows the performances on DTLZ2. After 670 million function evaluations, the naive NSGA-II was still not able to come close (with an error metric value of 0.01) to the true nadir point on the 10-objective DTLZ2 problem. However, the extremized-crowded NSGA-II took an average of 99,000 evaluations to achieve the task. Because of the computational inefficiencies associated with the naive NSGA-II approach, we did not perform any run for 15 or more objectives, whereas the extremized-crowded NSGA-II could find the nadir point up to the 20-objective DTLZ2 problem.

The nature of the plots for the extremized-crowded NSGA-II in both problems is found to be sub-linear on logarithmic axes. This indicates a lower than exponential scaling property of the proposed extremized-crowded NSGA-II. It is important to emphasize here that estimating the nadir point requires identification of the critical points. Since this requires that an evolutionary approach essentially puts its population members on the Pareto-optimal front, an adequate computational effort must be spent to achieve this task. However, results shown earlier for three to 10-objective problems have indicated that the computational effort needed by the extremized crowded NSGA-II approach is smaller when compared to the naive NSGA-II. It is worth pointing out here that decision makers do not necessarily want to or are not necessarily able to consider problems with very many objectives. However, the results of this study show a clear difference even with smaller problems involving, for example, five objectives.

6 Results of Tests with the Full Hybrid Nadir Point Estimation Procedure

Now, we apply the complete hybrid nadir point estimation procedure which makes a serial application of the extremized-crowded NSGA-II approach followed by the bilevel local search approach on three optimization problems. Since in the previous problems we identified difficulties with the worst-crowded NSGA-II, we do not employ the worst-crowded NSGA-II procedure here. The first two problems are numerical test problems taken from the MCDM literature on which the payoff table method is reported to have failed to estimate the nadir point accurately, and the third problem is a non-linear engineering design problem. All these problems adequately demonstrate the usefulness of the proposed hybrid procedure with the extremized-crowded NSGA-II approach. For all problems of this section, we use a population size of $20n$, where n is the number of variables and keep other NSGA-II parameters as they were used in the previous section. For both upper and lower-level optimizations in the local search, we have used the `fmincon` routine (implementing the sequential quadratic programming (SQP) method in which every approximated quadratic programming problem is solved using the BFGS quasi-Newton procedure) of MATLAB with default parameter values.

6.1 Problem KM

We consider a three-objective optimization problem, which provides difficulty for the payoff table method to estimate the nadir point. This problem was used in [26]:

$$\begin{array}{rl} \text{minimize} & \left\{ \begin{array}{l} -x_1 - x_2 + 5 \\ \frac{1}{5}(x_1^2 - 10x_1 + x_2^2 - 4x_2 + 11) \\ (5 - x_1)(x_2 - 11) \end{array} \right\}, \\ \text{subject to} & 3x_1 + x_2 - 12 \leq 0, \\ & 2x_1 + x_2 - 9 \leq 0, \\ & x_1 + 2x_2 - 12 \leq 0, \\ & 0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 6. \end{array} \tag{11}$$

Individual minimizations of objectives reveal the following three objective vectors: $(-2, 0, -18)^T$,

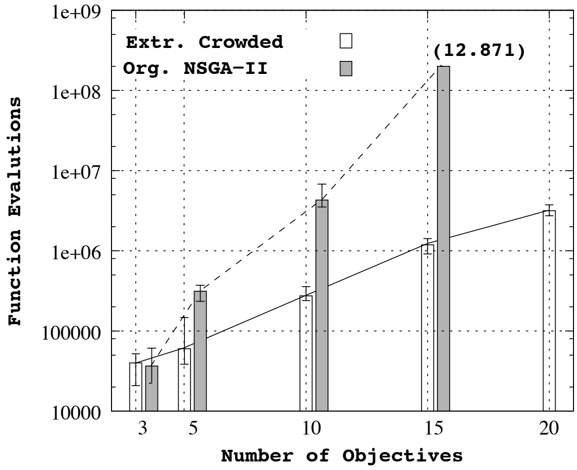

Figure 11: Function evaluations versus number of objectives for DTLZ1.

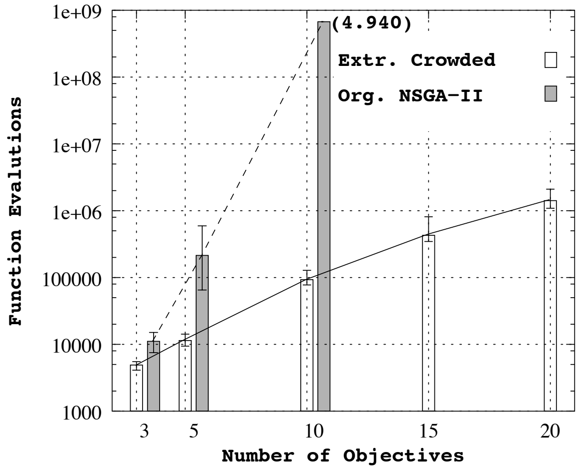

Figure 12: Function evaluations versus number of objectives for DTLZ2.

$(0, -3.1, -14.25)^T$ and $(5, 2.2, -55)^T$, thereby identifying the vector $\mathbf{z}^* = (-2, -3.1, -55)^T$ as the ideal objective vector. The payoff table method will then find $(5, 2.2, -14.25)^T$ as the estimated nadir point from these minimization results. Another study [22] used a grid-search strategy (computationally possible due to the presence of only two variables and three objectives) of creating a number of feasible solutions systematically and constructing the nadir point from the solutions obtained. The estimated nadir point was $(5, 4.6, -14.25)^T$ for this problem, which is different from that obtained from the payoff table. We now employ our nadir point estimation procedure to find the nadir point for this problem.

Step 1 of the procedure finds $\mathbf{z}^* = (-2, -3.1, -55)^T$ and $\mathbf{z}^w = (5, 4.6, -14.25)^T$ by minimizing and maximizing the objective functions individually.

In Step 2 of the procedure, we employ the extremized-crowded NSGA-II. As a result, we obtain four different non-dominated extreme solutions, as shown in the first column of Table 3. The extremized-crowded NSGA-II approach is terminated when the normalized distance metric does not change by an amount $\Delta = 0.0001$ in a consecutive $\tau = 50$ generations. It is interesting to note that the fourth solution is not needed to estimate the nadir point, but the extremized principle keeps this extreme solution corresponding to f_1 to possibly eliminate spurious solutions which may otherwise stay in the population and provide a wrong estimate of the nadir point (see Figure 3 for a discussion). Figure 13 shows the variation of the normalized distance metric value with generation, computed using the above-mentioned ideal and worst objective vectors. The NSGA-II procedure gets terminated at generation 135, due to the fall of the normalized distance value below the chosen threshold of 0.0001. At the end of Step 2, the estimated nadir point is $\mathbf{z}^{\rm nad} = (5, 4.6, -14.194)^T$, which seems to disagree on the third objective value with that found by the grid-search strategy.

To investigate if any further improvement is possible, we perform Step 3 and three times, each time starting with one of the first three solutions presented in Table 3, as they are the worst non-dominated solutions. The minimum and maximum objective vectors from these solutions are: $(-1, -3.1, -55)^T$ and $(5, 4.6, -14.194)^T$, respectively. Recall that the local search method suggested here is a bilevel optimization procedure in which the upper-level optimization uses a combination of a weight vector and a reference point as decision variable vector $(\mathbf{z}, \mathbf{w})$ with an objective of maximizing the objective value for which the corresponding NSGA-II solution is the worst. The lower-level optimization loop uses variable vector $\mathbf{x}$ and minimizes the corresponding achievement scalarizing function with $\rho = 10^{-5}$. To give an example, solution 1 from Table 3 corresponds to the worst value of the first objective. Thus, the upper-level optimization task maximizes objective f_1. The final three columns from the table show the outcome of the optimization run. Since this NSGA-II solution happens to already be the desired extreme solution, the upper-level optimization terminates after two iterations and declares the same solution as the outcome of the local search.

Table 3 clearly shows that solution 2 (the objective vector $(0.023, -3.100, -14.194)^T$, obtained by the extremized-crowded NSGA-II), was not a Pareto-optimal solution. The local search approach starting from this solution is able to find a better solution $(0, -3.1, -14.25)^T$. This shows the importance of em-

Table 3: extremized-crowded NSGA-II and local search method on Problem KM.

	$\mathbf{x}_{\text{NSGA-II}}$	Objective vector, $\mathbf{f}_{\text{NSGA-II}}$	$\mathbf{w}$	$\mathbf{z}$	Extreme point
1	$(0,0)^T$	$(5,2.2,-55)^T$	$(0.333,0.333,0.333)^T$	$(5,2.2,-55)^T$	$(5,2.2,-55)^T$
2	$(3.511,1.466)^T$	$(0.023,-3.100,-14.194)^T$	$(0.335,0.335,0.334)^T$	$(0.023,-3.085,-14.114)^T$	$(0,-3.1,-14.25)^T$
3	$(0,6)^T$	$(-1,4.6,-25)^T$	$(0.333,0.333,0.333)^T$	$(-1,4.6,-25)^T$	$(-1,4.6,-25)^T$
4	$(2.007,4.965)^T$	$(-1.973,-0.050,-18.060)^T$	Not worse in any objective, so not considered		

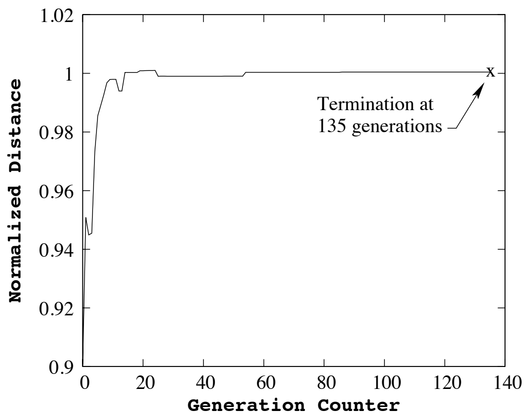

Figure 13: Variation of normalized distance metric with generation for problem KM.

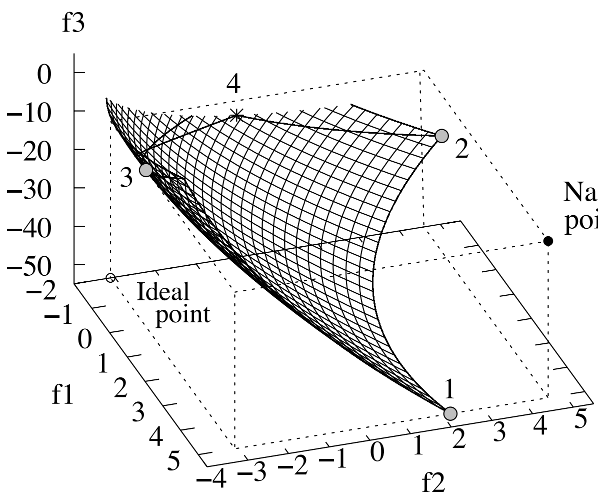

Figure 14: Pareto-optimal front with extreme points for problem KM. Point 4 is best for f_1, but not worst for any objective. Thus, it is redundant for estimating the nadir point.

ploying the local search approach in obtaining the exact nadir point. Because the extremized-crowded NSGA-II was able to find other two extreme solutions exactly, they could not be improved further by the local search procedure. Figure 14 shows the Pareto-optimal front for this problem. These three extreme Pareto-optimal points are marked on the front with a shaded circle. The fourth point is also shown with a star. Finally, in Step 4, the nadir point estimated by the combination of the extremized-crowded NSGA-II and local search is $(5, 4.6, -14.25)^T$, which is identical to that obtained by the grid search strategy [22]. As discussed earlier, the grid search strategy is not scalable to large problem sizes due to an exponential increase in computations.

The extremized-crowded NSGA-II approach took 5,440 solution evaluations and the three local search runs took a total of 1,583 solution evaluations, thereby requiring a total of 7,023 solution evaluations. Thus, the NSGA-II approach needed a major share of the overall computational effort of about 77% and the bilevel local search approach took only about 23% of the total effort.

6.2 Problem SW

Next, we consider another problem presented in [37]:

$$\begin{array}{rl} \text{minimize} & \left\{ \begin{array}{l} 9x_1 + 19.5x_2 + 7.5x_3 \\ 7x_1 + 20x_2 + 9x_3 \\ -(4x_1 + 5x_2 + 3x_3) \\ -(x_3) \end{array} \right\}, \\ \text{subject to} & 1.5x_1 - x_2 + 1.6x_3 \leq 9, \\ & x_1 + 2x_2 + x_3 \leq 10, \\ & x_i \geq 0, \quad i = 1,2,3. \end{array} \qquad (12)$$

The true nadir point for this problem is $\mathbf{z}^{\text{nad}} = (94.5, 96.3636, 0, 0)^T$. In [37], a close point $(94.4998, 95.8747, 0, 0)^T$ was found using multiple, bi-objective optimization runs. The estimation is different in its second objective value by about 0.5%. In the following, we show the results of our hybrid procedure.

In Step 1 of the procedure, we find the ideal and worst objectives values: $(0, 0, -31, -5.625)^T$ and $(97.5, 100, 0, 0)^T$, respectively. (These values are obtained by using the SQP routine of MATLAB.)

Thereafter, in Step 2, we apply the extremized-crowded NSGA-II procedure initializing the popu-

Table 4: Extremized-crowded NSGA-II and local search method on Problem SW.

	$\mathbf{x}_{\text{NSGA-II}}$	Objective vector, $\mathbf{f}_{\text{NSGA-II}}$	
1	$(0.0001, 0, 5.6249)^T$	$(42.1879, 50.6249, -16.8752, -5.6249)^T$	
2	$(0.0001, 3.1830, 3.6336)^T$	$(89.3219, 96.3635, -26.8164, -3.6336)^T$	
3	$(3.9980, 2.9998, 0.0003)^T$	$(94.4810, 87.9854, -30.9920, -0.0003)^T$	
4	$(0, 0, 0)^T$	$(0, 0, 0, 0)^T$	
	$\mathbf{w}$	$\mathbf{z}$	Extreme point
1	Not worse in any objective, so not considered		
2	$(1.0000, 0.9844, 0.7061, 0.8232)^T$	$(183.8020, 192.7266, -26.8004, -3.6336)^T$	$(89.3182, 96.3636, -26.8182, -3.6364)^T$
3	$(0.2958, 0.2540, 0.2006, 0.2486)^T$	$(188.9619, 184.3489, -30.9920, 5.6246)^T$	$(94.5000, 88.0000, -31.0000, 0.0000)^T$
4	$(0.25, 0.25, 0.25, 0.25)^T$	$(0, 0, 0, 0)^T$	$(0, 0, 0, 0)^T$

lation around $x_i \in [0, 10]$ for all three variables. The NSGA-II is terminated when the change in the normalized distance value in the past 50 generations is below the threshold of $\Delta = 0.0001$. Figure 15 shows the change in the normalized distance value with the generation counter and indicates that the NSGA-II run was terminated at generation 325. We obtain four different non-dominated solutions, which are tabulated in Table 4. The minimum and maximum objective vectors are: $(0.0000, 0.0000, -30.9920, -5.6249)^T$ and $(94.4810, 96.3635, 0.0000, 0.0000)^T$, respectively. Notice that this maximum vector is close to the true nadir point mentioned above. We shall now investigate whether the proposed local search is able to improve this point to find the nadir point more accurately.

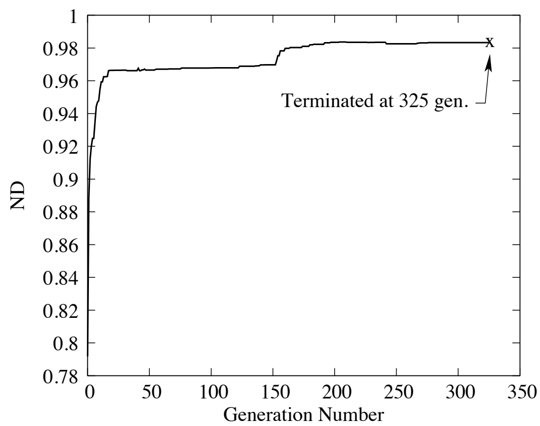

Figure 15: Variation of normalized distance metric with generation for problem SW.

We observe that the first solution does not correspond to the worst value for any objective. Thus, in Step 3, we employ the bilevel local search procedure only for the other three solutions. The resulting solutions and corresponding $\mathbf{z}$ and $\mathbf{w}$ vectors are shown in the table. For solutions 2 and 3, we maximize objectives f_2 and f_1, respectively. Since solution 4 is worst with respect to both objectives f_3 and f_4, we maximize a normalized composite objective: $-[(f_3(\mathbf{x}) - f_3^{\min})/(f_3^{\max} - f_3^{\min}) + (f_4(\mathbf{x}) - f_4^{\min})/(f_4^{\max} - f_4^{\min})]$, where maximum and minimum objective values are those obtained by the modified NSGA-II in Step 2.

From the obtained local search solutions (the last column in the table), in Step 4, we estimate the nadir point as $(94.5000, 96.3636, 0, 0)^T$, which is identical to the true nadir point for this problem. In this problem, the NSGA-II approach required 12,640 solution evaluations out of an overall 13,032 solution evaluations. Thus, the bilevel local search required only 392 solution evaluations (only about 3% of the overall effort). Thus, the use of the extremized-crowded NSGA-II allowed near critical points to be found by taking most of the computational effort and the use of the bilevel local search ensured finding the critical points by taking only a small fraction of the overall computational effort, despite the bilevel nature of the optimization procedure.

6.3 Welded Beam Design Optimization

So far, we have applied the hybrid nadir point estimation procedure to numerical test problems. They have given us confidence about the usability of our procedure. Next, we consider an engineering design problem related to a welded beam having three objectives, for which the exact nadir point is not known. In this problem, we compare our proposed nadir point estimation procedure with the naive NSGA-II approach for number of computations needed by each procedure and also to investigate whether an identical nadir point is estimated by each procedure.

This problem is well-studied [11, 35] having four design variables, $\mathbf{x} = (h, \ell, t, b)^T$ (dimensions specifying the welded beam). Minimizations of cost of fabrication, end deflection, and normal stress are of importance in this problem. There are four non-linear constraints involving shear stress, normal stress, a physical property, and buckling limitation. The mathematical description of the problem is given below:

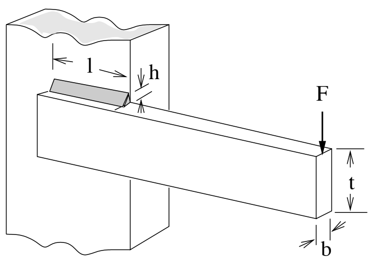

Figure 16: The welded beam design problem.

$$\begin{array}{ll} \text{minimize} & \left\{ \begin{array}{l} f_1(\mathbf{x}) = 1.10471h^2\ell + 0.04811tb(14.0+\ell) \\ f_2(\mathbf{x}) = \delta(\mathbf{x}) = 2.1952/t^3b \\ f_3(\mathbf{x}) = \sigma(\mathbf{x}) = 504,000/t^2b \end{array} \right\}, \\ \text{subject to} & g_1(\mathbf{x}) \equiv 13,600 - \tau(\mathbf{x}) \geq 0, \\ & g_2(\mathbf{x}) \equiv 30,000 - \sigma(\mathbf{x}) \geq 0, \\ & g_3(\mathbf{x}) \equiv b - h \geq 0, \\ & g_4(\mathbf{x}) \equiv P_c(\mathbf{x}) - 6,000 \geq 0, \\ & 0.125 \leq \ell, t \leq 10, \\ & 0.125 \leq h, b \leq 5, \end{array} \tag{13}$$

where the terms $\tau(\mathbf{x})$ and $P_c(\mathbf{x})$ are given as

$$\begin{array}{rcl} \tau(\mathbf{x}) & = & \left[(\tau'(\mathbf{x}))^2 + (\tau''(\mathbf{x}))^2 + \ell\tau'(\mathbf{x})\tau''(\mathbf{x})/ \right. \\ & & \left. \sqrt{0.25(\ell^2+(h+t)^2)}\right]^{1/2}, \\ P_c(\mathbf{x}) & = & 64,746.022(1-0.0282346t)tb^3, \end{array}$$

where

$$\begin{array}{rcl} \tau'(\mathbf{x}) & = & \dfrac{6,000}{\sqrt{2}h\ell}, \\ \tau''(\mathbf{x}) & = & \dfrac{6,000(14+0.5\ell)\sqrt{0.25(\ell^2+(h+t)^2)}}{2\left[0.707h\ell(\ell^2/12+0.25(h+t)^2)\right]}. \end{array}$$

In this problem, we have no knowledge on the ideal and worst objective values. Since these values will be required for computing the normalized distance metric value for terminating the extremized-crowded NSGA-II, we first find them here.

6.3.1 Step 1: Computing Ideal and Worst Objective Vectors

We minimize and maximize each of the three objectives to find the individual extreme points of the feasible objective space. For this purpose, we have used a single-objective real-parameter genetic algorithm with the SBX recombination and the polynomial mutation operators [12, 11]. We use a different set of parameter values from that of our multi-objective NSGA-II studies: population size = 100, maximum generations = 500, recombination probability = 0.9, mutation probability = 0.1, distribution index for recombination = 2, and distribution index for mutation = 20. These values are usually followed in other single-objective real-parameter GA studies [17, 9]. After a solution is obtained by a GA run, it is attempted to improve by a local search (LS) approach – the SQP procedure coded in MATLAB is applied with default parameter values to minimize individual objective functions in the feasible set. Table 5 shows the corresponding extreme objective values before and after the local search approaches. Interestingly, the use of the local search improves the cost objective from 2.3848 to 2.3810. As an outcome of the above single-objective optimization tasks, we obtain the ideal and worst objective values, as shown below:

	Cost	Deflection	Stress
Ideal	2.3810	0.000439	1008
Worst	333.9095	0.0713	30000

6.3.2 Step 2: Applying Extremized-Crowded NSGA-II

First, we apply the extremized-crowded NSGA-II approach with an identical parameter settings as used above, except that for the SBX recombination $\eta_c = 10$ is used, according to the recommendation in [11] for multi-objective optimization. The suggested termination criterion on the normalized distance (*ND*) metric is used with the above ideal and worst objective values. Figure 17 shows the variation of the *ND* metric with generation. It is interesting to note

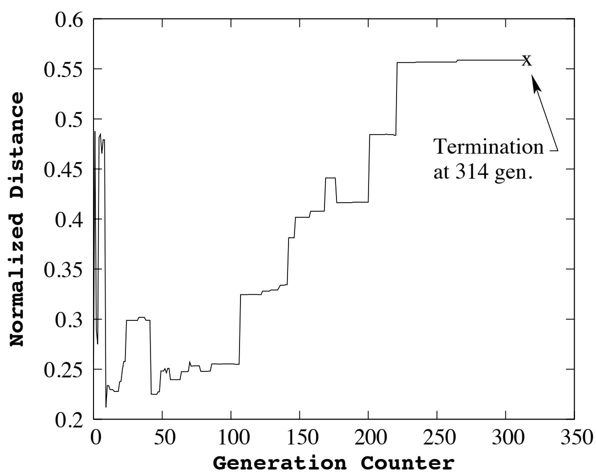

Figure 17: Variation of normalized distance metric with generation for the welded beam design problem.

how the normalized distance metric, starting from a small value (meaning that the estimated nadir point is closer to the worst objective vector), reaches a stabilized value of 0.5587. The NSGA-II procedure gets terminated at generation 314.

Table 5: Minimum and maximum objective values of three objectives. The values marked with a (*) for variables x_1 and x_2 can take other values without any change in the optimal objective value and without making the overall solution infeasible.

	Cost	Deflection	Stress	x_1	x_2	x_3	x_4
Minimum	2.3848			0.2428	6.2664	8.2972	0.2443
Min. after LS	2.3810			0.2444	6.2175	8.2915	0.2444
Maximum	333.9095			5	10	10	5
Max. after LS	333.9095			5	10	10	5
Minimum		0.000439		(*)4.4855	(*)9.5683	10	5
Min. after LS		0.000439		(*)4.4855	(*)9.5683	10	5
Maximum		0.0713		0.8071	5.0508	1.8330	5
Max. after LS		0.0713		0.8071	5.0508	1.8330	5
Minimum			1008	(*)4.5959	(*)9.9493	10	5
Min. after LS			1008	(*)4.5959	(*)9.9493	10	5
Maximum			30000	2.7294	5.7934	2.3255	3.1066
Max. after LS			30000	0.7301	5.0376	2.3308	3.0925

Table 6: Two population members obtained using the extremized crowded NSGA-II approach.

Sol. No.	Cost	Deflection	Stress	x_1	x_2	x_3	x_4
extremized-crowded NSGA-II							
1.	36.4347	0.000439	1008	1.5667	0.5389	10	5
2.	2.8235	0.0169	28088.3266	0.3401	4.6715	7.2396	0.3424
After local search							
1.	36.4209	0.000439	1008	1.7345	0.4789	10	5
2.	2.3810	0.0158	30000	0.2444	6.2175	8.2915	0.2444

Interestingly, only two non-dominated extreme points are found by the extremized-crowded NSGA-II. They are shown in Table 6. From these two solutions, the estimated nadir point after Step 2 is $(36.4347, 0.0169, 28088.3266)^T$. In a three-objective problem, the presence of only two extreme points signifies that two of the three objectives may be *correlated* to each other on the Pareto-optimal front. We shall discuss this aspect more later.

6.3.3 Step 3: Applying Local Search

The two solutions obtained are now attempted to be improved by the bilevel local search approach, one at a time. The minimum and maximum objective vectors obtained from the NSGA-II solutions (from Table 6) are as follows: $\mathbf{f}^{\min} = (2.8235, 0.000439, 1008)^T$ and $\mathbf{f}^{\max} = (36.4347, 0.0169, 28088.3266)^T$. Since the first solution corresponds to the worst of objective f_1, the upper-level loop of the local search for Solution 1 maximizes f_1. The resulting solution is shown in Table 6 under the heading 'After local search'. A slightly better solution is obtained using the local search.

For solution 2 of Table 6, objectives f_2 and f_3 are both worst. Thus, we maximize a normalized quantity arising from both objectives: $\sum_{i=2}^{3}(f_i(\mathbf{x}) - f_i^{\min})/(f_i^{\max} - f_i^{\min})$. The local search finds a non-dominated solution which seems to be better in terms of the first two objectives but worse in the third objective. The weight vector obtained by the upper-level loop of the local search is $(0.2470, 0.3333, 0.4196)^T$ and the corresponding reference point is $(2.8235, 0.0169, 55168.65)^T$. An investigation will reveal that the local search utilized a reference point which has identical values for the first two objectives and a much worse f_3 value than the NSGA-II solution. Then, by using a weight vector which has more or less equal value for all three objectives, the upper loop is able to locate the critical point corresponding to the second and third objectives. Interestingly, this critical point corresponds to the minimum f_1 value which is exactly the same as that obtained by the minimization of the cost objective alone in Table 5. It is clear that the extremized NSGA-II approach in Step 2 found a solution close to an extreme Pareto-optimal solution and the application of Step 3 helps to move this solution to the extreme Pareto-optimal solution.

Observing these two final solutions, in Step 4, we can now estimate the nadir point (cost, deflection, stress) for the welded beam design problem as

$$\text{Nadir point: } (36.4209, 0.0158, 30000)^T.$$

Note that this point is different from the worst objective vector of the entire feasible search space computed earlier. Out of a total of 31,551 solution evaluations, the bilevel local search required only 51 solution evaluations, thereby demanding a tiny fraction of 0.16% of the overall computational effort.

6.3.4 Comparison with the Naive NSGA-II Approach

We now apply the naive NSGA-II approach to the same problem to investigate whether an identical nadir point is obtained. In the naive approach, we first generate a set of Pareto-optimal points by a combination of the original NSGA-II and a local search approach. The range of the Pareto-optimal front, thus found, will provide us information about the nadir point of the problem. We use an identical parameter setting as used in the extremized-crowded NSGA-II run. The local search approach used here is applied to NSGA-II solutions one at a time and is described in Chapter 9.6 of [11]. We employed `fconmin` routine of MATLAB for this purpose. In Figure 18, we show the NSGA-II solutions with circles and their

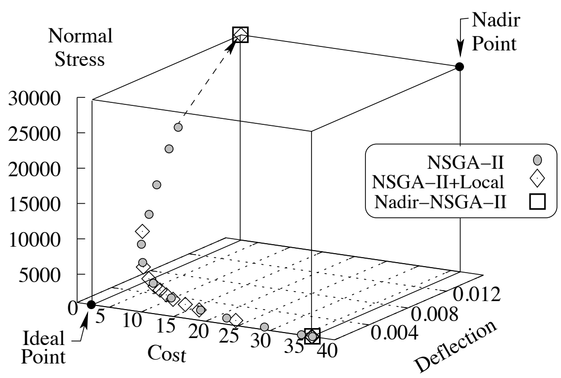

Figure 18: Pareto-optimal front and estimation of the nadir point.

improvements by the local search method with diamonds. Two non-dominated extreme solutions obtained using our nadir point estimation procedure are marked using squares. Both approaches find an identical nadir point, thereby providing confidence to our approach proposed. However, the overall function evaluations needed to complete the naive NSGA-II and local searches for obtaining the distributed set of Pareto-optimal points was 102,267, compared to a total of 31,551 function evaluations needed with our proposed nadir point estimation procedure. For a four-variable, three-objective problem, a reduction of about 70% computations with our proposed approach to find an identical nadir point is a significant achievement.

It is also interesting to note that despite the use of three objectives, the Pareto-optimal front is one-dimensional in this problem. If the obtained front is projected on the deflection-stress (f_2-f_3) plane, it can be seen that these two objectives are correlated to each other. Therefore, in addition to finding the nadir point, the number of extreme solutions $\mathbf{x}_{\text{EA}}$ found by the extremized-crowded NSGA-II procedure may provide ideas about the dimensionality of the Pareto-optimal front – an added benefit which can be obtained by performing the nadir point estimation task before attempting to solve a problem for multiple Pareto-optimal solutions. A significant amount of research efforts is now being made in handling many-objective problems using evolutionary algorithms and in automatically identifying redundant objectives in a problem [18, 4, 24]. An analysis of critical points obtained by the proposed extremized crowded NSGA-II procedure for identifying possible redundancy in objectives is worth pursuing further and remains as a viable approach in this direction.

7 Discussions and Extensions

In this paper, we have combined the flexibility in an EMO search with an ingenious local search procedure. By redirecting the focus of an EMO's diversity-preserving operator towards the extreme non-dominated solutions, we have suggested an extremized-crowded NSGA-II procedure which is able to find representative points close to extreme points of the Pareto-optimal front, not only to three or four-objective problems, but to as many as 20-objective problems. By proposing a bilevel local search procedure of choosing an appropriate reference point near an obtained NSGA-II solution and a suitable weight vector for finding the critical point corresponding to the worst non-dominated solutions obtained by the NSGA-II procedure, we have demonstrated the working of the hybrid procedure to a number of challenging test and practical optimization problems.

To make NSGA-II's search more efficient, a *mating restriction* strategy can be added so that a better stability of multiple extreme solutions is maintained in the population. Restricting recombination among neighboring solutions in the objective space may also allow a focused search, thereby finding a better approximation of extreme solutions. For this purpose, the emphasis for extreme solutions can also be implemented on other EMO procedures, such as SPEA2 [39] or PESA [8] or others.

In the upper-level local search approach (problem (7)), the upper bound on the reference point $\mathbf{z}$ is chosen rather conservatively. Since the task is to perform a local search, a tighter and more problem-informatic upper bound, such as a more relaxed bound on the worst objective value and a more restricted bound on the other objectives can be used for a computationally faster procedure. Similarly, the bounds on the weight vector can also be chosen with some problem information derived from the location of the particular NSGA-II solution vis-a-vis other solutions. In fact, the inclusion of the weight vector $\mathbf{w}$ as a part of the decision variable vector of the upper-level optimization may not be needed. By fixing the weight vector based on the location of the NSGA-II solution, the upper-level optimization may be used to find an optimal $\mathbf{z}$ corresponding to the extreme Pareto-optimal solution. This task may require less computational effort due to the reduction in decision variables on the upper-level optimization loop.

In another approach, the bilevel local search procedure suggested here can be integrated within the NSGA-II procedure as an additional operator. The local search can be applied to a few selected solutions of a NSGA-II population after every few generations. This on-line procedure will guarantee finding (locally) Pareto-optimal solutions whenever the local search is applied. However, the computational advantage, if any, compared to the proposed hybrid approach of this study will be an interesting future research worth pursuing.

8 Conclusions

We have proposed a hybrid methodology involving evolutionary and local search approaches to address an age-old yet open research issue of estimating the nadir point accurately in a multi-objective optimization problem. By definition, a nadir point is constructed from the worst objective values corresponding to the solutions of the Pareto-optimal front. It has been argued that the estimation of the nadir point is an important task in multi-objective optimization. Since the nadir point relates to the critical *Pareto-optimal* points, the estimation of a nadir point is also a difficult and challenging task. Since intermediate Pareto-optimal solutions are not important in this task, the suggested modified NSGA-II approaches have emphasized their search for finding the worst or extreme solutions corresponding to each objective. To enhance the convergence properties and make the approaches reliable, the modified NSGA-II approaches have been combined with a reference point based bilevel local search approach. The upper-level search uses a combination of a reference point and a weight vector as a variable vector, which is then evaluated by using a lower-level search of solving the corresponding achievement scalarizing function. While the lower-level search ensures converging to a local Pareto-optimal solution, the upper-level search drives the procedure to converge to the critical point of an objective function.

The extremized-crowded approach has been found to be capable of making a quicker estimate of the nadir point than a naive approach (of employing the original NSGA-II approach to first find a set of non-dominated solutions and then construct the nadir point) on a number of benchmark problems having three to 20 objectives and on other problems including a difficult engineering design problem involving non-linear objectives and constraints. Emphasizing solutions corresponding to the extreme objective values on a non-dominated front has been found to be a better approach than emphasizing solutions having the worst objective values alone. Since the former approach maintains a diverse set of solutions near both best and worst objective values, thereby not allowing spurious dominated solutions to remain in the population, the result of the search is better and more reliable than that of the worst-crowded approach.

The computational effort to estimate the nadir point has been observed to be much smaller (more than an order of magnitude) for benchmark test problems having a large number of objectives than the naive NSGA-II approach. Moreover, since the extremized-crowded NSGA-II approach has been able to find solutions close to the critical points, the local search procedure has been found to take only a fraction of the overall computational effort. Thus, the bilevel nature of the proposed local search procedure does not seem to affect much the overall computational effort of the hybrid approach.

Despite the algorithmic challenge posed by the task of estimating the nadir point in a multi-objective optimization problem, in this paper, we have listed a number of reasons for which nadir objective vectors are useful in practice. They included normalizing objective functions, giving information about the ranges of objective functions within the Pareto-optimal front to the decision maker, visualizing Pareto-optimal solutions, and enabling the decision maker to use different interactive methods. What is common to all these is that the nadir objective vector can be computed beforehand, without involving the decision maker. Thus, it is not a problem if several hundred function evaluations are needed in the extremized-crowded NSGA-II in most problems. Approximating the nadir point can be an independent task to be executed before performing any decision analysis.

One of the reasons why it may be advisable to use some interactive method for identifying the most preferred solution instead of trying to approximate the whole set of Pareto-optimal solutions is that for problems with several objectives, for example, the NSGA-II approach requires a huge number of evaluations to find a representative set. For such problems, the nadir point may be estimated quickly and reliably using the proposed hybrid NSGA-II-cum-local-search procedure. The extremized-crowded NSGA-II approach can be applied with a coarse termination requirement, so as to obtain near extreme non-dominated solutions quickly. Then, the suggested local search approach can be employed to converge to the extreme Pareto-optimal solutions reliably and accurately. Thereafter, an interactive procedure (like NIMBUS [30, 33, 32], for example) (using both ideal and nadir points obtained) can be applied interactively with a decision-maker to find a desired Pareto-optimal solution as the most preferred solution.

This study is important in another aspect, as well. The proposed nadir point estimation procedure uses a hybridization of EMO and a local search based MCDM approach. The population aspect of EMO has been used to find near extreme non-dominated solutions simultaneously and the reference point based local search methodology helped converge to true extreme Pareto-optimal solutions so that the nadir point can be estimated reliably and accurately. Such collaborative EMO-MCDM studies may help develop efficient hybrid procedures which use best aspects of both contemporary fields of multi-objective optimization. Hopefully, this study will motivate researchers to engage in more such collaborative studies for the benefit of either field and, above all, to the triumph of the field of multi-objective optimization.

Acknowledgments

Authors acknowledge the FiDiPro support of the first author by the Academy of Finland under grant 119319 and the Foundation of Helsinki School of Economics. The work of the second author was partly supported by the Jenny and Antti Wihuri Foundation, Finland.

References

[1] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Laritchev. Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, 1(3):366–375, 1971.

[2] H. P. Benson. Existence of efficient solutions for vector maximization problems. *Journal of Optimization Theory and Applications*, 26(4):569–580, 1978.

[3] J. Branke, K. Deb, K. Miettinen, and R. Slowinski. *Multiobjective optimization: Interactive and evolutionary approaches*. Springer-Verlag, Berlin, Germany, 2008.

[4] D. Brockhoff and E. Zitzler. Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In K. H. Waldmann and U. M. Stocker, editors, *Operations Research Proceedings 2006*, pages 423–429. Springer, 2007.

[5] J. T. Buchanan. A naive approach for solving MCDM problems: The GUESS method. *Journal of the Operational Research Society*, 48(2):202–206, 1997.

[6] C. A. C. Coello, D. A. VanVeldhuizen, and G. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer, 2002.

[7] D. W. Corne and J. D. Knowles. Techniques for highly multiobjective optimisation: some non-dominated points are better than others. In *GECCO'07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 773–780, New York, NY, USA, 2007. ACM Press.

[8] D. W. Corne, J. D. Knowles, and M. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature VI (PPSN-VI)*, pages 839–848, 2000.

[9] K. Deb. Mechanical component design using genetic algorithms. In D. Dasgupta and Z. Michalewicz, editors, *Evolutionary Algorithms in Engineering Applications*, pages 495–512. New York: Springer-Verlag, 1997.

[10] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation Journal*, 7(3):205–230, 1999.

[11] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.

[12] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[14] K. Deb and S. Chaudhuri. I-MODE: An interactive multi-objective optimization and decision-making using evolutionary methods. In *Proceedings of Fourth International Conference on Evolutionary Multi-Criteria Optimization (EMO 2007)*, pages 788–802, 2007.

[15] K. Deb, S. Chaudhuri, and K. Miettinen. Towards estimating nadir objective vector using evolutionary approaches. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, pages 643–650. New York: The Association of Computing Machinery (ACM), 2006.

[16] K. Deb and D. E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, 1989.

[17] K. Deb and M. Goyal. A robust optimization procedure for mechanical component design based on genetic adaptive search. *Transactions of the ASME: Journal of Mechanical Design*, 120(2):162–164, 1998.

[18] K. Deb and D. Saxena. Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, pages 3352–3360, 2006.

[19] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multi-objective Optimization*, pages 105–145. London: Springer-Verlag, 2005.

[20] M. I. Dessouky, M. Ghiassi, and W. J. Davis. Estimates of the minimum nondominated criterion values in multiple-criteria decision-making. *Engineering Costs and Production Economics*, 10:95–104, 1986.

[21] M. Ehrgott and D. Tenfelde-Podehl. Computation of ideal and nadir values and implications for their use in MCDM methods. *European Journal of Operational Research*, 151:119–139, 2003.

[22] P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto Navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, in press.

[23] H. Isermann and R. E. Steuer. Computational experience concerning payoff tables and minimum criterion values over the efficient set. *European Journal of Operational Research*, 33(1):91–97, 1988.

[24] A. L. Jaimes, C. A. C. Coello, and D. Chakraborty. Objective reduction using a feature selection technique. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2008)*, pages 673–680, 2008.

[25] V. Khare, X. Yao, and K. Deb. Performance scaling of multi-objective evolutionary algorithms. In *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632)*, pages 376–390, 2003.

[26] K. Klamroth and K. Miettinen. Integrating approximation and interactive decision making in multicriteria optimization. *Operations Research*, 56:222–234, 2008.

[27] P. Korhonen, S. Salo, and R. Steuer. A heuristic for estimating nadir criterion values in multiple objective linear programming. *Operations Research*, 45(5):751–757, 1997.

[28] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.

[29] M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz. Three different ways for incorporating preference information in interactive reference point based methods. Technical Report W-410, Helsinki School of Economics, Helsinki, Finland, 2006.

[30] K. Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer, Boston, 1999.

[31] K. Miettinen. Graphical illustration of Pareto optimal solutions. In T. Tanino, T. Tanaka, and M. Inuiguchi, editors, *Multi-Objective Programming and Goal Programming: Theory and Applications*, pages 197–202. Springer-Verlag, Berlin, Heidelberg, 2003.

[32] K. Miettinen and M. M. Mäkelä. Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. *Optimization*, 34(3):231–246, 1995.

[33] K. Miettinen and M. M. Mäkelä. Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170(3):909–922, 2006.

[34] K. Miettinen, M. M. Mäkelä, and K. Kaario. Experiments with classification-based scalarizing functions in interactive multiobjective optimization. *European Journal of Operational Research*, 175(2):931–947, 2006.

[35] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell. *Engineering Optimization Methods and Applications*. New York : Wiley, 1983.

[36] K. Sindhya, K. Deb, and K. Miettinen. A local search based evolutionary multi-objective optimization technique for fast and accurate convergence. In *Proceedings of the Paralle Problem Solving From Nature (PPSN-2008)*. Berlin, Germany: Springer-Verlag, 2008.

[37] M. Szczepanski and A. P. Wierzbicki. Application of multiple crieterion evolutionary algorithm to vector optimization, decision support and reference point approaches. *Journal of Telecommunications and Information Technology*, 3:16–33, 2003.

[38] A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Applications*, pages 468–486. Berlin: Springer-Verlag, 1980.

[39] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2001. International Center for Numerical Methods in Engineering (CIMNE).